

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 6



**TÍTULO:** Plataforma de Gestión de Ayudas para  
Sistemas de Información Geográfica

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS**

**AUTOR:** Adrian Bello Mesa.

**TUTOR:** Ing. Yoenis Pantoja Zaldívar

La Habana, Junio 2012  
“Año 54 de la Revolución”

# DECLARACIÓN DE AUTORÍA

---

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad 6 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año 2012.

---

Adrian Bello Mesa

---

Ing. Yoenis Pantoja Zaldívar

# DATOS DE CONTACTO

---

## DATOS DE CONTACTO

**Tutor:** Ing. Yoenis Pantoja Zaldívar

Jefe del Departamento de Técnicas de Programación. Facultad 6. Universidad de las Ciencias Informáticas. Graduado de Informática en el año 2006 en la Universidad de Holguín.

Datos del contacto: [ypantojaz@uci.cu](mailto:ypantojaz@uci.cu)

# AGRADECIMIENTOS

---

## AGRADECIMIENTOS

*Uno siempre esta rodeado de personas que protagonizan o forman parte de la historia particular de cada cual, personas que brindan equilibrio y que dejan su propia huella en el librito de todos nosotros y que hay que darles crédito y agradecer su participación en esos buenos o malos momentos que conforman nuestra vida.*

*¡Gracias! Antes que a nada en el mundo, a las personas que mas quiero, y que me dieron la vida, a mis padres, ellos son el modelo que llevo la vida intentando seguir, los que como nadie han sufrido y disfrutado cada uno de esos momentos, los que se han entregado en vida para que sus hijos sean hoy lo que son, dos profesionales listos para empezar su propia vida. Gracias a mi hermanita linda que siempre me saca de los apuros y a la que le toca cargar con su hermano, y lo que le falta todavía. Fruto de ese amor que nuestros padres siempre han cultivado en nosotros.*

*Gracias, muchas gracias, a mi familia que me ha dado todo su apoyo y no ha estado al margen de mi carrera y mi vida ni un segundo, la familia perfecta de la que no puedo estar más orgulloso porque es imposible.*

*Gracias, al regalito de mi vida, a mi muchachita bonita, mi calmante, mi aire, mi todo, mi novia y futura esposa Yanelis, el amor de mi vida, mi grillito de la conciencia que nunca me dejo vacilar y desviarme de mi objetivo, gracias mi amor.*

*Gracias a mi familia de San Cristobal que han estado pendiente de todo, y me han apoyado siempre.*

## AGRADECIMIENTOS

---

*Gracias, a todos los que han estado en todo momento a mi lado sufriendo y disfrutando la universidad en todas sus etapas en estos seis años, mis colegas y amigos, mis compadres de misión y de batalla que al fin cumplimos el objetivo. Ya somos ingenieros.*

*Gracias a Esther, Esperanza, Librado, Yolexis y Roberto que siempre me ayudaron y apoyaron incondicionalmente durante mi carrera.*

*Gracias a las personas que me ayudaron en toda la investigación, a mi cuñado Reinier por su incondicional y desinteresado apoyo, a Mailin por su labor de cotutora, a la gente de CISE, a mi tutor Yoenis por su labor orientadora y todos los que de una forma u otra colaboraron con mi tarea La Tesis Mundial.*

*Gracias a todas aquellas personas que contribuyeron en mi formación profesional, mis profesores, compañeros de aula y de estudios.*

*Gracias a Fidel por permitirme continuar algo de lo que el comenzó.*

*En fin gracias a todas esas personas que tuvieron que ver con el éxito que hoy ostento, gracias a todos aquellos que de una manera u otra dejaron su huella en ese librito que se va escribiendo a diario, el mío.*

# DEDICATORIA

---

## DEDICATORIA

*Le dedico el presente Trabajo de Diploma a mis padres, mi confianza e inspiración, ellos tienen todo el mérito de mi éxito.*

# RESUMEN

---

## RESUMEN

En la Universidad de las Ciencias Informáticas (UCI) se está llevando a cabo un proceso de desarrollo de Sistemas de Información Geográfica (SIG) con el objetivo de suplir la necesidad que tiene el país de resolver problemas de planificación y gestión, además de adentrarse en el mercado mundial y la comercialización de este tipo de soluciones informáticas. Este proceso trae consigo la demanda de software aplicativos que aumenten la funcionalidad, usabilidad y calidad de los mismos.

Como motivo de la investigación, se fundamenta la dificultad que existe en el centro de desarrollo de software GEYSED de la facultad 6 de la UCI, a la hora de desarrollar e implementar sistemas de ayudas para SIG, en entornos Web y con tecnologías libres. Con el objetivo de contribuir al entendimiento del objeto fundamental de la investigación y al desarrollo de la misma, se realiza un estudio del comportamiento de este tema en las comunidades de desarrollo de estos tiempos y de las principales herramientas usadas actualmente para estos fines. Además se justifica el uso de la metodología de desarrollo XP para guiar la concepción de la solución propuesta en este trabajo. Como resultado de todo el proceso investigativo e ingenieril se obtiene una aplicación Web que permite la confección de sistemas de ayudas para SIG de forma ágil y estándar.

**Palabras Claves:** Ayuda, Metodología Plataforma, SIG, y Sistema.

# ÍNDICE

---

## ÍNDICE

DECLARACIÓN DE AUTORÍA .....	II
DATOS DE CONTACTO .....	III
AGRADECIMIENTOS .....	IV
DEDICATORIA.....	VI
RESUMEN .....	VII
INTRODUCCIÓN.....	13
CAPÍTULO I: Fundamentación Teórica.....	16
1.1. <i>Introducción.</i> .....	16
1.2. <i>Conceptos Asociados al Dominio del Problema.</i> .....	16
1.2.1. <i>Sistemas de Ayuda.</i> .....	17
1.2.2. <i>Importancia de los Sistemas de Ayudas.</i> .....	17
1.2.3. <i>Tipos de Sistemas de Ayuda.</i> .....	18
1.3. <i>Software Gestores o Generadores de Ayudas.</i> .....	20
1.3.1. <i>RoboHelp.</i> .....	20
1.3.2. <i>WinHelp.</i> .....	22
1.3.3. <i>WebHelp.</i> .....	23
1.3.4. <i>HelpMagician.</i> .....	23
1.3.5. <i>Otros Generadores de Ayuda.</i> .....	24
1.4. <i>Conclusiones.</i> .....	24
CAPÍTULO 2: Investigación y Análisis de las Tecnologías Libres para el Desarrollo de Sistemas Gestores de Ayuda para SIG.....	25
2.1. <i>Introducción.</i> .....	25
2.2. <i>Sistema Gestor de Bases de Datos.</i> .....	25
<i>PostgreSQL.</i> .....	25
2.3. <i>Servidor Web Apache.</i> .....	26
2.4. <i>Lenguajes de Programación.</i> .....	27
2.4.1. <i>Hypertext Preprocessor (PHP)</i> .....	27
2.4.2. <i>Java Script.</i> .....	28
2.5. <i>Metodologías de Desarrollo.</i> .....	29
2.5.1. <i>Extreme Programming (XP).</i> .....	30
2.6. <i>Javascript Extendido (Ext JS).</i> .....	33



# ÍNDICE

---

2.7. Entorno de Desarrollo Integrado (IDE) .....	34
Netbeans .....	34
2.8. Framework.....	35
CodeIgniter.....	35
2.9. Conclusiones.....	36
<b>CAPÍTULO 3: Presentación de la Solución Propuesta. ....</b>	<b>37</b>
3.1. Introducción.....	37
3.2. Propuesta del Sistema.....	37
3.3. Personas Relacionadas con el Sistema .....	37
3.4. Fase de Exploración .....	37
3.4.1. Requisitos no Funcionales .....	38
3.4.2. Historias de Usuario.....	38
3.4.3. Historias de Usuario Definidas.....	40
3.5. Planificación.....	47
3.5.1. Estimación de Esfuerzos.....	48
3.5.2. Plan de Iteraciones.....	49
3.5.3. Plan de duración de las Iteraciones .....	50
3.5.4. Plan de Entrega .....	51
3.6. Conclusiones.....	51
<b>CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta. ....</b>	<b>52</b>
4.1. Introducción.....	52
4.2. Diseño .....	52
4.2.1. Tarjetas CRC .....	53
4.2.2. Arquitectura del Sistema .....	54
4.3. Implementación.....	57
4.3.1. Iteración # 1.....	57
4.3.2. Iteración # 3.....	59
4.3.3. Iteración # 4.....	60
4.3.4. Iteración # 5.....	61
4.3.5. Iteración # 6.....	62
4.3.6. Iteración # 7.....	62
4.4. Estándar de Codificación.....	64
4.5. Pruebas.....	65
4.5.1. Pruebas Unitarias.....	65

# ÍNDICE

---

4.5.2. Pruebas de Aceptación.....	66
4.6. Conclusiones.....	73
CONCLUSIONES.....	74
RECOMENDACIONES.....	75
BIBLIOGRAFÍA CONSULTADA.....	76
BIBLIOGRAFÍA CITADA.....	78
ANEXOS.....	80

## ÍNDICE DE TABLAS

Tabla # 1 Personas Relacionadas con el Sistema.....	37
Tabla # 2 Historia de Usuario.....	40
Tabla # 3 HU Crear Proyecto.....	40
Tabla # 4 HU Editar Proyecto.....	41
Tabla # 5 HU Eliminar Proyecto.....	41
Tabla # 6 Adicionar Ficha.....	42
Tabla # 7 HU Editar Ficha.....	42
Tabla # 8 HU Eliminar Ficha.....	43
Tabla # 9 HU Adicionar Contenido.....	43
Tabla # 10 HU Editar Contenido.....	44
Tabla # 11 Eliminar Contenido.....	44
Tabla # 12 Previsualizar Proyecto.....	45
Tabla # 13 HU Importar Proyecto.....	45
Tabla # 14 HU Adicionar Glosario.....	46
Tabla # 15 HU Editar Glosario.....	46
Tabla # 16 HU Eliminar Glosario.....	47
Tabla # 17 HU Autenticar.....	47
Tabla # 18 Estimación de Esfuerzos.....	49
Tabla # 19 Plan de Duración de las Iteraciones.....	50
Tabla # 20 Plan de Entrega.....	51
Tabla # 21 Tarjeta CRC GestionarAyudaController.....	54
Tabla # 22 Tarjeta CRC GestionarAyudaModel.....	54

# ÍNDICE

---

Tabla # 23 HUs Iteración 1. ....	57
Tabla # 24 Tarea de Ingeniería # 1, HU # 1, Iteración # 1. ....	58
Tabla # 25 Tarea de Ingeniería # 1, HU # 2, Iteración # 1. ....	58
Tabla # 26 Tarea de Ingeniería # 1, HU # 3, Iteración # 1. ....	58
Tabla # 27 HUs Iteración # 3. ....	59
Tabla # 28 Tarea de Ingeniería # 1, HU # 7, Iteración # 3. ....	59
Tabla # 29 Tarea de Ingeniería # 1, HU # 8, Iteración # 3. ....	59
Tabla # 30 Tarea de Ingeniería # 1, HU # 9, Iteración # 3. ....	60
Tabla # 31 HUs Iteración # 4. ....	60
Tabla # 32 Tarea de Ingeniería # 1, HU # 10, Iteración # 4. ....	60
Tabla # 33 Tarea de Ingeniería # 1, HU # 11, Iteración # 4. ....	61
Tabla # 34 Tarea de Ingeniería # 1, HU # 12, Iteración # 4. ....	61
Tabla # 35 HUs Iteración # 5. ....	61
Tabla # 36 Tarea de Ingeniería # 1, HU # 13, Iteración # 5. ....	62
Tabla # 37 HUs Iteración # 6. ....	62
Tabla # 38 Tarea de Ingeniería # 1, HU # 14, Iteración # 6. ....	62
Tabla # 39 HUs Iteración # 7. ....	62
Tabla # 40 Tarea de Ingeniería # 1, HU # 15, Iteración # 7. ....	63
Tabla # 41 Tarea de Ingeniería # 1, HU # 16, Iteración # 7. ....	63
Tabla # 42 Tarea de Ingeniería # 1, HU # 17, Iteración # 7. ....	63
Tabla # 43 Tarea de Ingeniería # 1, HU # 18, Iteración # 7. ....	64
Tabla # 44 Caso de Prueba # 1 para HU Crear Proyecto. ....	67
Tabla # 45 Caso de Prueba # 1 para HU Editar Proyecto. ....	67
Tabla # 46 Caso de Prueba # 1 para HU Eliminar Proyecto. ....	68
Tabla # 47 Caso de Prueba # 1 para HU Adicionar Ficha. ....	68
Tabla # 48 Caso de Prueba # 1 para HU Editar Ficha. ....	69
Tabla # 49 Caso de Prueba # 1 para HU Eliminar Ficha. ....	69
Tabla # 50 Caso de Prueba # 1 para HU Adicionar Contenido. ....	70
Tabla # 51 Caso de Prueba # 1 para HU Editar Contenido. ....	70
Tabla # 52 Caso de Prueba # 1 para HU Eliminar Contenido. ....	70
Tabla # 53 Caso de Prueba # 1 para HU Previsualizar Proyecto. ....	71

# ÍNDICE

---

Tabla # 54 Caso de Prueba # 1 para HU Exportar Proyecto. ....	71
Tabla # 55 Caso de Prueba # 1 para HU Adicionar Glosario.....	72
Tabla # 56 Caso de Prueba # 1 para HU Editar Glosario. ....	72
Tabla # 57 Caso de Prueba # 1 para HU Eliminar Glosario.....	73
Tabla # 58 Caso de Prueba # 1 para HU Autenticar.....	73
Tabla # 59 HU Adicionar Categoría. ....	80
Tabla # 60 HU Editar Categoría.....	81
Tabla # 61 HU Eliminar Categoría. ....	81
Tabla # 62 HUs Iteración # 2 .....	81
Tabla # 63 Tarea de Ingeniería # 1, HU # 4, Iteración # 2. ....	82
Tabla # 64 Tarea de Ingeniería # 1, HU # 5, Iteración # 2. ....	82
Tabla # 65 Tarea de Ingeniería # 1, HU # 5, Iteración # 2. ....	82
Tabla # 66 Caso de Prueba # 1 para HU Adicionar Categoría. ....	83
Tabla # 67 Caso de Prueba # 1 para HU Editar Categoría.....	83
Tabla # 68 Caso de Prueba # 1 para HU Eliminar Categoría. ....	84
Tabla # 69 Caso de Prueba # 1 para HU Adicionar Categoría con Conflicto.....	84
Tabla # 70 Caso de Prueba # 1 para HU Eliminar Categoría con Conflicto.....	85
Tabla # 71 Caso de Prueba # 1 para HU Adicionar Contenido con Conflicto. ....	85
Tabla # 72 Caso de Prueba # 1 para HU Exportar Proyecto con Conflicto.....	86

## INTRODUCCIÓN

La Universidad de las Ciencias Informáticas cuenta con una línea de producción considerable destinada a la elaboración de SIG. Como todo proceso de desarrollo de software en el contexto ingenieril está guiado y orientado por una metodología de desarrollo, díganse, Proceso Unificado Racional (RUP por su acrónimo en inglés [Rational Unified Process]) y Programación Extrema (XP acrónimo en inglés de [Extreme Programming]) por solo citar algunas de la más usadas por la comunidad de desarrolladores de estos tiempos. Según este tipo de metodologías, en el levantamiento de requisitos (en específico los no funcionales) y a la hora de confeccionar el Manual de Usuario, aspectos que en gran medida se encargan de garantizar la imagen, navegabilidad y usabilidad del software entre otros parámetros para su confección, se acota la importancia de que todo software cuente con un Sistema de Ayuda.

Una de las ventajas más importantes que un programador de aplicaciones puede ofrecer a los usuarios es un sistema de ayuda eficaz, éste será su referencia cuando estén confundidos o desorientados. Estos sistemas siempre han acompañado al software, y en la actualidad la mayoría de los programas ofrecidos en el mercado vienen acompañados por este importante componente, ya que sirve de guía al usuario a la hora de utilizar el programa que acompañan, describiéndolo, explicando la forma de uso, o bien argumentando los conceptos que maneja.

Atendiendo a este requisito tan necesario y por la importancia que tiene para la satisfacción del usuario para el que esté diseñada la aplicación, contribuyendo al máximo aprovechamiento y cumplimiento del objetivo para la que esta fue creada, en la Facultad 6 de la UCI en el Centro de Geoinformática y Señales Digitales (GEYSED) se realizó un minucioso estudio de lo que demandaba un proceso productivo como el que se está llevando a cabo, caracterizado por el desarrollo masivo de SIG de disímiles temáticas.

En el transcurso de este proceso salió a colación la complejidad en el mismo, además de haber morosidad y descentralización a la hora de implementar sistemas de ayuda para un mismo tipo de software, caracterizado por poseer muchas funcionalidades prácticamente básicas y con un alto por ciento de similitud, independientemente del tema que trate cada SIG en cuestión. Además resulta muy tedioso para los desarrolladores implementar un sistema de ayuda para cada SIG en específico, provocando demoras e incumplimientos en el plan de entrega de los productos.

# INTRODUCCIÓN

---

Por otra parte a pesar de la infinidad de soluciones informáticas que hoy circulan en la red de redes y en el mercado internacional para estos fines, la gran mayoría de ellas son herramientas privativas, impedimento este que, por la situación económica y política del país, pone freno al desarrollo de la industria del software en Cuba, además estas herramientas están desarrolladas con un fin muy general.

Teniendo en cuenta la situación planteada surge para la presente investigación el siguiente **problema a resolver**: *¿cómo estandarizar el proceso de desarrollo de Sistemas de Ayudas en el Departamento Geoinformática del Centro GEYSED en las diferentes soluciones desarrolladas?*

Se define como **objeto de estudio** los *Procesos de desarrollo de Sistemas de Ayudas para SIG*. De ello se deriva que el **campo de acción** donde se enmarca la investigación sea *la automatización del proceso de desarrollo de Sistemas de Ayudas para SIG*.

Se plantea como **objetivo general** *Desarrollar una plataforma de gestión de ayudas para Sistemas de Información Geográfica en entornos Web para el Departamento Geoinformática del Centro GEYSED utilizando tecnologías libres*.

Para darle cumplimiento al objetivo planteado se definieron las diferentes **tareas de la investigación** que se relacionan a continuación:

1. Caracterizar el estado del arte de las soluciones informáticas para el desarrollo de ayudas.
2. Seleccionar y argumentar la Metodología de Desarrollo de Software a usar en el proceso.
3. Identificar y fundamentar las tecnologías necesarias para el desarrollo del sistema.
4. Identificar las principales funcionalidades de la herramienta.
5. Analizar y diseñar el sistema.
6. Implementar las funcionalidades del sistema.
7. Validar el resultado obtenido con la personalización de la herramienta para un SIG específico.

El cumplimiento exitoso de las tareas expuestas anteriormente contribuye a la verificación de la siguiente **idea a defender**: *desarrollando una plataforma de gestión de Sistemas de Ayudas para SIG se logrará la estandarización en el proceso de desarrollo de ayudas para estos sistemas*.

En el desarrollo de la investigación se utilizan varios métodos científicos, tanto teóricos como empíricos.

# INTRODUCCIÓN

---

Dentro de los Métodos Teóricos son utilizados:

1. Histórico – Lógico: Se aplica para realizar el seguimiento de la evolución del Proceso de Desarrollo de Sistemas de Ayudas para SIG y una predicción de lo que puede hacerse en el futuro, en este caso se utiliza para investigar la existencia de aplicaciones que gestionen ayudas o sistemas de ayudas para SIG.
2. Modelación: Se utiliza para representar por medio de diagramas el proceso de gestión de ayudas referente a los SIG, teniendo como resultado un mejor entendimiento de la posible solución a implementar. Para ello se utiliza la modelación teórica, la cual tiene la capacidad de representar las características y relaciones fundamentales del fenómeno, proporcionar explicaciones y servir como guía para generar hipótesis teóricas.
3. Análisis y Síntesis: Se utiliza para identificar y analizar las diversas funcionalidades de los sistemas de ayudas convencionales que pueden ser aplicadas en la gestión de sistemas de ayudas y su posterior síntesis a partir de la selección de las funcionalidades aplicables a los procesos de gestión de ayudas de acuerdo a las necesidades específicas de cada SIG.

## CAPÍTULO I: Fundamentación Teórica.

### **1.1. Introducción.**

En este capítulo son tratados temas y conceptos con relación al dominio del problema en cuestión, los mismos facilitan el entendimiento del contenido, y la consiguiente familiarización con la temática abordada. Se desglosa y fundamenta el estado del arte en consecuencia con el objeto de estudio, el campo de acción de la investigación, así como otras particularidades referentes a esta última.

### **1.2. Conceptos Asociados al Dominio del Problema.**

Con el objetivo de lograr un adecuado nivel de comprensión de los temas que son abordados en este capítulo, se describen a continuación un grupo de conceptos relacionados con el dominio del problema, entre los que se destacan: Sistemas de Ayuda, Sistemas de Información Geográfica, Gestión, Plataforma, Entorno Libre.

**Gestión:** Conjunto de trámites, acciones, diligencias, tareas que se ejecutan para darle solución a una problemática o lograr un objetivo.

**Plataforma:** Es un conjunto de tecnologías para construir sitios o aplicaciones Web, plataforma es precisamente el software base sobre el cual se van a desarrollar estas aplicaciones.

**Sistema de Ayuda:** Conjunto de programas diseñados para proporcionar asistencia al usuario sobre una aplicación o un sistema informático. **(Kearsley, 1987)**

**Sistema de Información Geográfica:** Un Sistema de Información Geográfica (SIG o GIS, en su acrónimo inglés [Geographic Information System]) es una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión. También puede definirse como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestre y construido para satisfacer necesidades concretas de información. En el sentido más estricto, es cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. En un sentido más genérico, los SIG son herramientas que permiten a los usuarios crear consultas



interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones (**Chang, 2006**).

**Software Libre:** El software libre (en inglés free software, esta denominación también se confunde a veces con gratis por la ambigüedad del término en el idioma inglés) es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente. Según la Free Software Foundation, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado. (**Öztürk, 2002**)

## **1.2.1. Sistemas de Ayuda.**

El desarrollo de sistemas de ayuda es un campo prácticamente novedoso en el desarrollo de aplicaciones informáticas, sus inicios están enmarcados en la segunda mitad de la década de los 70. Sin embargo, ahora la mayor parte de las aplicaciones incluyen algún tipo de ayuda. En teoría, con la mejora de las interfaces y del diseño de programas, actualmente sería posible construir aplicaciones tan sencillas de utilizar que el usuario no necesitase ningún tipo de ayuda adicional. Sin embargo, circunstancias como la creciente complejidad de los sistemas informáticos, su continua evolución y su uso generalizado por usuarios no expertos, hacen que los sistemas de ayuda sean un componente fundamental para simplificar el uso y el aprendizaje de los entornos informáticos.

En los últimos años se han publicado numerosos trabajos que versan sobre los Sistemas de Ayuda, lo que indica que se trata de un campo de trabajo muy activo, aunque a pesar de ello el marco de estos sistemas no está claramente definido (Kearsley, 2004). La carencia de una definición precisa se debe principalmente a dos razones: las diferentes formas de entender lo que es la ayuda informatizada y las diferentes áreas de aplicación. Los Sistemas de Ayuda se aplican en muy variados campos, proporcionando ayuda a los usuarios en situaciones muy distintas. Así, por ejemplo, existen sistemas de ayuda tanto para las instalaciones industriales complejas como para la navegación aérea o la toma de decisiones bancarias. Dado el marco de la aplicación, la investigación se centra en los sistemas de ayuda utilizados para proporcionar asistencia a los usuarios de aplicaciones informáticas de uso general, enfocándose en aquellas que son desarrolladas en entornos Web con tecnologías libres.

## **1.2.2. Importancia de los Sistemas de Ayudas.**

Uno de los objetivos ideales de cualquier aplicación software bien diseñada sería tener una estructura tan evidente y una interfaz tan comprensible para el usuario, que no fuera necesario proporcionar

# CAPÍTULO I: Fundamentación Teórica

---

ningún tipo de ayuda adicional para su perfecto uso. No obstante, en general, en las aplicaciones dicho objetivo está lejos de cumplirse, por lo que se hace necesario proporcionar asistencia a los usuarios. La mejor forma de proporcionar dicha asistencia es mediante la incorporación de Sistemas de Ayuda en las aplicaciones. **(Cigarrán, 1996)**

Varias son las nuevas circunstancias que hacen que aumente la necesidad de ayuda por parte de los usuarios. Por una parte, como se ha comentado, los programas ofrecen cada vez mayores funcionalidades, permitiendo realizar operaciones más complejas y sofisticadas. Por otra parte, y de forma simultánea, el número de aplicaciones software va creciendo y su uso se generaliza, introduciendo a nuevas clases de usuarios en el mundo informático.

Las computadoras ya no sólo son utilizadas por expertos con una formación técnica y una sólida base informática, sino también por otras personas, a menudo con escasos conocimientos de los conceptos básicos y con una mínima o nula experiencia. Muchas de las aplicaciones se utilizan de forma ocasional, o durante cortos períodos de tiempo, para satisfacer necesidades concretas (por ejemplo, la generación de presentaciones o la creación de gráficos en una hoja de cálculo, sin que se llegue a dominar su uso.

Además de esa utilización ocasional, también es frecuente el cambio de un programa a otro dentro de la misma área de aplicación, como por ejemplo el cambio de un programa de edición de textos a uno de otra índole, así como la evolución a nuevas versiones del mismo programa que ofrecen nuevas capacidades (siendo habitual que se haya modificado la interfaz). En estos casos, los usuarios intentan aplicar su conocimiento anterior y, a pesar de su experiencia, es muy probable que se encuentren con problemas en su uso y necesiten ayuda. **(Duffy, 1989)**

Para solucionar estos problemas, la mejor alternativa se encuentra en la potenciación y mejora de los Sistemas de Ayuda. Mediante estos se puede integrar el uso y el aprendizaje de los programas, de modo que sea posible su utilización a partir de conocimientos mínimos y se proporcione asistencia al usuario cuando éste encuentre problemas que no sabe resolver. **(Carroll, 1988)**

### ***1.2.3. Tipos de Sistemas de Ayuda.***

Un Sistema de Ayuda interactiva para una aplicación informática es un programa diseñado para asistir al usuario en el uso de la aplicación. Es decir, su propósito principal es el de proporcionar una respuesta inmediata a los problemas específicos que se presentan en la utilización de la aplicación

**(Duffy, 1992)**. Por otra parte a partir de los años 80 surgieron nuevos tipos de Sistemas de Ayuda que basan su aporte de ayuda en la utilización de técnicas de inteligencia artificial y en la inclusión de modelos explícitos, tanto del dominio como del usuario. Son los que se denominan Sistemas de Ayuda Inteligente. **(Wilensky, 1984)**

### **1.2.3.1. Sistema de Ayuda Interactiva.**

Los programas que proporcionan algún tipo de ayuda interactiva a los usuarios de las aplicaciones informáticas se han venido denominando de diversas formas y han utilizado enfoques muy diferentes. Entre otros, se han empleado términos como Sistema de Ayuda, asistente, Sistema de Ayuda Inteligente, documentación interactiva, entrenador, consejero, interfaz inteligente e incluso combinaciones de esas palabras **(Duffy, 1992)**. Cada una de estas denominaciones en menor medida tienen un significado diferente pero todas tienen el mismo fin que las asemeja y es que todas tienen como objetivo proveer al usuario de la asistencia necesaria ante cualquier eventualidad.

Se utiliza el término “advisory interface” para referirse genéricamente a los entrenadores<sup>1</sup>, a la documentación de referencia, a la ayuda interactiva y a otros tipos de aplicaciones que proporcionan soporte a los usuarios. **(Carroll, 1988)**

Se denominan Sistemas de Ayuda Interactivos a los programas que ayudan a los usuarios de computadoras a interiorizar la tecnología de modo que se mejore el rendimiento. **(Duffy, 1989)**

Un Sistema de Ayuda Interactiva es aquel que provee al usuario de toda la información acerca de cómo usar la aplicación en el momento que éste la requiera de forma manual y permita una especie de comunicación estrecha entre aplicación y usuario.

### **1.2.3.2. Sistemas de Ayuda Inteligente.**

Un Sistema de Ayuda Inteligente implica un sistema que pueda tomar la iniciativa para proveer ayuda que no se ha solicitado. Un ejemplo es el proyecto Eurohelp, donde se define un asistente inteligente como aquel que captura la interacción del usuario con la aplicación, siendo capaz de responder preguntas y de irrumpir para ayudar al usuario cuando hay problemas o cuando se presenta una oportunidad adecuada para enseñar al usuario alguna nueva funcionalidad del entorno. **(Winkels, 1992)**

---

<sup>1</sup>Sistemas Entrenadores: aquellos encargados de tutorial al usuario en la forma en que desempeñan una determinada tarea o la manera en que usan un software de manera interactiva.

Aunque hay ciertas discrepancias sobre lo que se puede considerar como ayuda inteligente, se usa el término con un significado amplio, definiendo un Sistema de Ayuda Inteligente como aquel que tiene capacidades de adaptación a las circunstancias concretas en las que se solicita la ayuda. **(Shank, 1991)** Esta capacidad de adaptación para las ayudas inteligentes también es destacada como factor discriminante por otros autores como Greg Kearsley. No obstante, esa capacidad de adaptación y de adecuación de la ayuda proporcionada puede ser enfocada de forma muy variada, tanto en su concepción como en su implementación final.

Por ejemplo, podemos encontrar desde el disponer de información diferente para cada tipo de usuario hasta una adaptación automática del comportamiento del programa al usuario concreto para seleccionar la información que se considere más adecuada (incluso realizándose modificaciones de la interfaz, lo que se podría clasificar como interfaz inteligente, que no son más que aquellas que tratan de entender de forma activa qué es lo que el usuario intenta hacer e integran distintas ayudas dentro de la interfaz **(Duffy, 1989)**. Para conseguir este comportamiento adaptativo resulta clave la utilización de una representación explícita de las características de los usuarios.

Un Sistema de Ayuda Inteligente es aquel que es capaz de anticiparse a las necesidades del usuario en dependencia de las características individuales del mismo, además de brindar información que pueda servir de ayuda al usuario de manera automática.

### **1.3. Software Gestores o Generadores de Ayudas.**

En la actualidad en el mundo del desarrollo de software existen un conjunto de herramientas para la creación de ayudas, estas herramientas son denominadas Generadores de Sistemas de Ayudas. Entre ellas existen familias de generadores que se han desarrollado al ritmo que la industria del software lo ha pedido, de esta manera ya existen ayudas para aplicaciones Web específicamente, por citar un ejemplo. A continuación se relacionan algunos de los más usados.

#### **1.3.1. RoboHelp.**

Es un producto para la creación de sistemas de ayuda y documentación profesionales para aplicaciones de escritorio y Web, tales como aplicaciones .NET y aplicaciones dinámicas de Internet. Podrá crear sistemas de ayuda que incluyan elementos tales como temas de ayuda, tablas de materias, índices, glosarios y ayuda contextual entre otras características. Éste tiene una variedad de productos entre los que se encuentran RoboHelp Office, RoboHelp Office Pro, RoboHelp Office Pro for .NET, RoboHelp para Word, tal y como se relaciona a continuación.

## **RoboHelp Office.**

Es el estándar en lo que se refiere a la autoría de sistemas de ayuda. Con RoboHelp Office puede trabajar en el editor de HTML incorporado de RoboHelp, en cualquier editor popular, tal como Dreamweaver, o en Microsoft Word, tienen una gran flexibilidad pues le permite generar cualquier formato de ayuda popular, crear sistemas que se ejecuten con cualquier explorador y cualquier plataforma, crear documentos listos para enviar a la imprenta, además de personalizar el contenido para diversos tipos de usuarios. **(Adobe, 2012)**

### **Requisitos que debe cumplir su PC:**

- Pentium 300 MHz o más rápido.
- Sistema operativo de 32 bits - Windows XP, ME, 2000, 98, o NT 4.0\*.
- 128 MB de RAM (se recomiendan 256 MB).
- MS Word 2003, 2002, 2000, 97 para la autoría en un entorno basado en MS Word\*.
- Internet Explorer 5.0 o posterior.
- 200 MB de espacio disponible en el disco.

## **RoboHelp Office Pro.**

RoboHelp Office Pro incluye todas las características de RoboHelp Office además de una poderosa funcionalidad basada en el servidor, incluido el seguimiento en tiempo real de la actividad del usuario final en el Sistema de Ayuda y la aplicación. RoboHelp Office Pro para .NET contiene todas las características de RoboHelp Office Pro, además de otras características para el desarrollo de sistemas de ayuda para Microsoft .NET. **(Adobe, 2012)**

### **Requisitos que debe cumplir su PC:**

- Procesador Pentium III de 450MHz o más rápido.
- 128 MB de RAM.
- 50 MB de espacio disponible en el disco.
- 50 MB de espacio disponible en el disco después de la instalación (para un rendimiento óptimo).
- Microsoft Windows NT 4.00 Server con IIS 4.0 o Microsoft Windows 2000 Server con IIS 5.0.

## **RoboHelp Office Pro for .NET.**

Tiene todas las características de RoboHelp Office Pro, además de otras características para el desarrollo de sistemas de ayuda que se despliegan en Microsoft .NET. Éste también permite añadir búsquedas en lenguaje natural, hacer búsquedas en documentos externos, incluye también ayuda automatizada sin conexión. **(Adobe, 2012)**

### **Requisitos que debe cumplir su PC**

- Procesador Pentium III de 450MHz o más rápido.
- 128 MB de RAM.
- 50 MB de espacio disponible en el disco.
- 50 MB de espacio disponible en el disco después de la instalación (para un rendimiento óptimo).
- Microsoft Windows NT 4.00 Server con IIS 4.0 o Microsoft Windows 2000 Server con IIS 5.0.

## **RoboHelp para Word.**

Se diseñó para los usuarios que prefieren trabajar en Microsoft Word que es su editor predeterminado, y que le permite al usuario abrir, crear y modificar los documentos que van a conformar la ayuda o el curso en línea. **(Adobe, 2012)**

### **Requisitos que debe cumplir su PC**

- Pentium 300 MHz o más rápido.
- 128 MB de RAM (se recomiendan 256 MB).
- Sistema operativo de 32 bits - Windows XP, ME, 2000, 98, o NT 4.0\*.
- Microsoft Word 2002, 2000, 97\*.
- Internet Explorer 5.0 o posterior.
- 200 MB de espacio disponible en el disco.

### **1.3.2. WinHelp.**

Es un formato propietario para crear documentación de WinHelp.hip de manera rápida y eficiente. La interfaz simple del procesador de textos hace fácil el acceso a sus potentes características. Fue desarrollado por Microsoft para proporcionar la ayuda ("Ayuda de Microsoft Windows") en el entorno Windows 95 y posteriores. Se pueden generar ficheros compatibles con este formato (.HLP). El visor

presenta tres visiones (Tripane): la tabla de contenidos (TOC), el índice de términos y una utilidad de búsqueda. La empresa Blue SkySoft ha desarrollado un producto compatible WINHELP<sub>2000</sub> - ROBOHELP.

WinHelp también apoya una característica conocida como ayuda sensible al contexto. La ayuda sensible al contexto es la ayuda que es apropiada a donde está el usuario en la aplicación informática, y a lo que están intentando hacer. Una característica crítica de la seguridad es algo que se puede también incluir un archivo del DLL que contiene código de encargo y que lo asocia a los asuntos de WinHelp. Con eficacia esto hace archivos de .HLP equivalentes a los ejecutables. **(Microsoft, 2012)**

### **1.3.3. WebHelp.**

Formato de entrega de ayuda desarrollado por Help Corporation, los creadores de RoboHelp. Es un método para la entrega de ayuda en línea o la documentación en un explorador Web. WebHelp es simplemente un conjunto de archivos HTML y Java. Su formato flexible se puede utilizar:

- Para ofrecer ayuda para los sitios Web, intranets y aplicaciones basadas en Web.
- Como el Sistema de Ayuda para el software tradicional, sin importar el lenguaje de programación utilizado para desarrollar.
- Para entregar un conjunto completo de documentación en una plataforma y formato de navegador Web independiente.

Un sistema de WebHelp es básicamente un documento HTML basado en marcos. El marco izquierdo contiene un applet de Java que funciona como la tabla de contenido o de navegación para el sistema. El cuadro de la derecha contiene el contenido real. **(Adobe, 2012)**

### **1.3.4. HelpMagician.**

Es una herramienta de autoría de ayuda para los archivos de ayuda de Windows. Permite crear archivos de ayuda para sus aplicaciones, Sistema de Ayuda HTML, catálogos en línea, páginas Web y cualquier otro tipo de documentos de hipertexto. Contiene todas las características del procesador de textos estándar: corrector ortográfico, funciones reemplazar estilos y apartado de búsqueda. Funciona en todas las versiones de 32 bits de Windows (95, 98, NT, ME, 2000, XP) y con todos los lenguajes de programación.

Brinda la posibilidad de incluir los controles de formulario HTML, los controles ActiveX, Java Controls, Java Script, VBScript, navegación y botones de navegación en la parte superior de las páginas HTML. Es un asistente de ayuda interactiva, que ofrece todos los pasos para crear un archivo de ayuda de manera sencilla. Incluye una utilidad de captura de pantalla, una Ayuda de Visual Basic, un mapa de bits para convertir fuentes en mapas de bits y una herramienta de importación RTF. En el mundo de la autoría de sistemas de ayuda HelpMagician es muy popular cuando se necesita crear ayudas HTML o páginas Web. **(Softlockup, 2009)**

### **1.3.5. Otros Generadores de Ayuda.**

Hoy día existen una amplia gama de software cuya función específica es la generación y diseño de sistemas de ayuda entre los que se encuentran:

- HTML Help Workshop 1.4 SDK.
- Help y Manual 4.1.
- DocBuilder 2.2.
- AnetHelpTool 6.5.0.3.
- Fast-Help 4.1.

No existen soluciones informáticas específicamente para el desarrollo de Sistemas Gestores de Ayuda para SIG de forma particular. La idea inicial del proyecto fue totalmente concebida por desarrolladores pertenecientes a la Factoría de Aplicativos para SIG del la Facultad 6 de la UCI en vistas a perfeccionar el proceso de desarrollo de SIG en la misma.

### **1.4. Conclusiones.**

Se puede concluir que en Cuba el proceso de desarrollo de SIG es bastante limitado, solo algunas empresas se dedican a implementar algunos componentes que manejan y gestionan información geográfica. A nivel mundial esta temática está bastante desarrollada y es de vital importancia para el país introducirse en el mercado del software como partidaria del uso y diseminación del Software Libre en la implementación de aplicaciones Web robustas. Se puede afirmar que en el mundo aunque no se desarrollan Sistemas de Ayuda para SIG de forma particular, existen infinidad de aplicaciones destinadas para el desarrollo de Sistemas de Ayuda o gestores de los mismos, algunos de los cuales se han referido en este capítulo, pero para Cuba resulta difícil el uso de éstas, porque la gran mayoría son propietarias, situación que afecta en gran medida la industria del software cubano.



## CAPÍTULO 2: Investigación y Análisis de las Tecnologías Libres para el Desarrollo de Sistemas de Ayuda para SIG

---

### CAPÍTULO 2: Investigación y Análisis de las Tecnologías Libres para el Desarrollo de Sistemas Gestores de Ayuda para SIG.

#### **2.1. Introducción.**

Para llevar a cabo cualquier proceso productivo en el que los Sistemas de Ayuda sean la razón de ser del mismo, es primordial la investigación y análisis de las tendencias de desarrollo existentes en el mundo, así como las herramientas y tecnologías inmersas en ello. Dentro de éstas, las libres son las más usadas, puesto que permiten la creación de sus componentes en la utilización de código abierto. En este capítulo se muestra una breve panorámica sobre las herramientas usadas en el desarrollo de estas aplicaciones.

#### **2.2. Sistema Gestor de Bases de Datos.**

Los sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. **(Henst, 2008)** La bibliografía especializada a menudo se refiere a estos sistemas como SGBD o DBMS, siendo ambos equivalentes y acrónimos de Sistema Gestor de Bases de Datos y Data Base Management System, respectivamente.

#### **PostgreSQL.**

PostgreSQL es un Sistema de Administración de Bases de Datos de Objetos Relacionales. Forma parte del movimiento de software libre y fue lanzado bajo la licencia BSD (Berkeley Software Distribution). Postgres ofrece una alternativa a los demás sistemas de administración de bases de datos. PostgreSQL no está controlado por una sola compañía, sino que cuenta con comunidad global de desarrolladores y compañías para su evolución, posee una serie de características positivas respecto a otros. Entre éstas cabe mencionar que es escalable, ajustable al número de procesadores y a la cantidad de memoria que posee el sistema de la forma más óptima. Por este motivo es capaz de soportar una mayor cantidad de peticiones simultáneamente.

Otra característica relevante de este gestor es la capacidad de almacenar procedimientos (funciones) en la propia base de datos, además de ser multiusuario, multiprogramado, con arquitectura cliente-servidor y control de privilegios de acceso; posee tipos internos que han sido mejorados, incluyendo

## CAPÍTULO 2: Investigación y Análisis de las Tecnologías Libres para el Desarrollo de Sistemas de Ayuda para SIG

---

nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales, además incluye características que permiten la orientación a objetos, incluye características de cómo puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

Es un Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos. Con cerca de una década de desarrollo, PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario, contando también con un amplio conjunto de enlaces con lenguajes de programación. **(Fabien Potencier, 2008)**

En el desarrollo de la aplicación se usa PostgreSQL por su carácter de software libre, posee características que permiten la Programación Orientada a Objetos (POO) y por su adaptabilidad a la sintaxis SQL. Además particularmente la arquitectura del proyecto aplicativos de GEYSED para el cual está diseñada la aplicación propone el uso del mismo como estrategia de compatibilidad y estandarización en el proceso de desarrollo.

### **2.3. Servidor Web Apache.**

El Servidor Web es el programa encargado de gestionar las peticiones de los usuarios que visitan su página Web. Es el programa encargado de mostrar sus páginas cuando un visitante realiza una petición en su dominio. Teniendo en cuenta que el 80% de las operaciones que realiza un usuario en el servidor, es visionar páginas Web, la elección del programa encargado para tal cometido es clave para el buen funcionamiento general del servidor. Apache es considerado el Servidor Web por excelencia, no sólo por su gran aceptación, puesto que casi el 70% de los servidores Internet confían en él, sino porque desde su nacimiento ha demostrado con creces su estabilidad, solidez y su mayor rendimiento ante sus competidores.

Porqué usar Apache:

- ✓ Fiabilidad: Alrededor del 90% de los servidores con más alta disponibilidad funcionan bajo servidores Apache.
- ✓ Software Libre: Apache es totalmente libre, y se distribuye bajo la licencia Apache Software License, que permite la modificación del código.

## CAPÍTULO 2: Investigación y Análisis de las Tecnologías Libres para el Desarrollo de Sistemas de Ayuda para SIG

---

- ✓ Extensibilidad: Se pueden añadir módulos para ampliar las ya de por sí amplias capacidades de Apache.

También dispone de una amplia variedad de módulos, que permiten desde generar contenido dinámico (con PHP, Java, Perl, Python,...), monitorizar el rendimiento del servidor, atender peticiones encriptados por SSL, crear servidores virtuales por IP o por nombre (varias direcciones Web son manejadas en un mismo servidor) y limitar el ancho de banda para cada uno de ellos. Dichos módulos están disponibles junto con su código fuente, por lo cual pueden ser incluso modificados por cualquier persona con conocimientos de programación.

### **2.4. Lenguajes de Programación.**

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente.

#### **2.4.1. Hypertext Preprocessor (PHP).**

PHP es un lenguaje de programación interpretado que permite la creación de páginas Web dinámicas. Se utiliza como código servidor aprovechando la programación orientada a objetos que implementa, sus características principales son la independencia de plataforma y su gratuidad. Otros lenguajes muy utilizados son ASP o JSP, que tienen características similares.

Este lenguaje fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. El mismo está preparado para realizar muchos tipos de aplicaciones Web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos.

**(Álvarez, 2007)**

#### Características de PHP:

- ✓ Muy sencillo de aprender.
- ✓ Similar en sintaxis a C y a PERL.
- ✓ Soporta en cierta medida la orientación a objeto. Clases y herencia.
- ✓ El análisis léxico para recoger las variables que se pasan en la dirección lo hace PHP de forma automática. Librándose el usuario de tener que separar las variables y sus valores.

## CAPÍTULO 2: Investigación y Análisis de las Tecnologías Libres para el Desarrollo de Sistemas de Ayuda para SIG

---

- ✓ Se puede incrustar código PHP con etiquetas HTML.
- ✓ Excelente soporte de acceso a base de datos.
- ✓ La comprobación de que los parámetros son válidos se hace en el servidor y no en el cliente (como se hace con Javascript) de forma que se puede evitar chequear que no se reciban solicitudes adulteradas.
- ✓ Además PHP viene equipado con un conjunto de funciones de seguridad que previenen la inserción de órdenes dentro de una solicitud de datos.
- ✓ Se puede hacer de todo lo que se pueda transmitir por vía HTTP.

### **2.4.2. Java Script.**

Javascript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página Web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario.

El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador, Javascript es el siguiente paso, después del HTML, que puede dar un programador de la Web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica.

Entre las acciones típicas que se pueden realizar en Javascript tenemos dos vertientes. Por un lado los efectos especiales sobre páginas Web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, Javascript permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

## CAPÍTULO 2: Investigación y Análisis de las Tecnologías Libres para el Desarrollo de Sistemas de Ayuda para SIG

---

Javascript es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Además, Javascript pone a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente. (Álvarez, 2007)

### Características de Javascript:

- ✓ Es simple, no hace falta tener conocimientos de programación para poder hacer un programa en JavaScript.
- ✓ Maneja objetos dentro de nuestra página Web y sobre ese objeto podemos definir diferentes eventos. Dichos objetos facilitan la programación de páginas interactivas, a la vez que se evita la posibilidad de ejecutar comandos que puedan ser peligrosos para la máquina del usuario, tales como formateo de unidades, modificar archivos etc.
- ✓ Es dinámico, responde a eventos en tiempo real. Eventos como presionar un botón, pasar el puntero del mouse sobre un determinado texto o el simple hecho de cargar la página o caducar un tiempo. Con esto podemos cambiar totalmente el aspecto de nuestra página al gusto del usuario, evitando tener en el servidor una página para cada gusto, hacer cálculos en base a variables cuyo valor es determinado por el usuario, etc.
- ✓ Javascript es manejado por eventos, puede responder a eventos como el movimiento del mouse y la carga de una página Web.
- ✓ Es independiente de cualquier plataforma. Los programas de javascript están diseñados para ejecutarse dentro de documentos HTML. Son independientes de cualquier plataforma o sistema operativo.
- ✓ Permite desarrollo rápido. El navegador Web y el código HTML manejan la mayoría de las características como formas, cuadros, y otros elementos de Interfaz Gráfica del Usuario (GUI). Esto hace que los programadores de Javascript no tienen que preocuparse de crear o manejar estos elementos en sus aplicaciones.
- ✓ Javascript es fácil de aprender: No incluye complejas reglas sintácticas.

### **2.5. Metodologías de Desarrollo.**

En la actualidad, es prácticamente imposible omitir la utilización de metodologías para el desarrollo y elaboración de aplicaciones ordenadas, por tanto, seguir metodologías y estándares conllevan a estar en competitividad en todo momento.

## CAPÍTULO 2: Investigación y Análisis de las Tecnologías Libres para el Desarrollo de Sistemas de Ayuda para SIG

---

### 2.5.1. Extreme Programming (XP).

XP es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo, cuyo plazo de entrega era ayer. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto. **(Sanchez, 2004)**

#### Características de XP:

- ✓ Pruebas Unitarias: Se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándose en algo hacia el futuro, se puedan hacer pruebas de las fallas que pudieran ocurrir.
- ✓ Refabricación: Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- ✓ Programación en pares: Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa.

#### ¿Qué es lo que propone XP?

- ✓ Empieza en pequeño y añade funcionalidad con retroalimentación continua.
- ✓ El manejo del cambio se convierte en parte sustantiva del proceso.
- ✓ El costo del cambio no depende de la fase o etapa.
- ✓ No introduce funcionalidades antes que sean necesarias.
- ✓ El cliente o el usuario se convierte en miembro del equipo.

#### Derechos del Cliente

- ✓ Decidir qué se implementa.
- ✓ Saber el estado real y el progreso del proyecto.
- ✓ Añadir, cambiar o quitar requerimientos en cualquier momento.
- ✓ Obtener lo máximo de cada semana de trabajo.
- ✓ Obtener un sistema funcionando cada 3 o 4 meses.

#### Derechos del Desarrollador

- ✓ Decidir cómo se implementan los procesos.

## CAPÍTULO 2: Investigación y Análisis de las Tecnologías Libres para el Desarrollo de Sistemas de Ayuda para SIG

---

- ✓ Crear el sistema con la mejor calidad posible.
- ✓ Pedir al cliente en cualquier momento aclaraciones de los requerimientos.
- ✓ Estimar el esfuerzo para implementar el sistema.
- ✓ Cambiar los requerimientos en base a nuevos descubrimientos.

Lo fundamental en este tipo de metodología es:

- ✓ La comunicación, entre los usuarios y los desarrolladores.
- ✓ La simplicidad, al desarrollar y codificar los módulos del sistema.
- ✓ La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales. **(Sanchez, 2004)**

Los "12 mandamientos" del XP:

- ✓ Programación 1: diseñar y codificar de forma simple.
  - para producir software sencillo de cambiar y de mantener.
- ✓ Programación 2: reorganizar el código frecuentemente.
  - para que éste tenga siempre y en todo momento el diseño más óptimo posible.
- ✓ Programación 3: fijar estándares de codificación.
  - para que el código resulte claramente legible por todos.
- ✓ Programación 4: fijar un vocabulario estándar para todos
  - para poder comunicar claramente las ideas.
- ✓ Desarrollo 1: todo el proceso es dirigido por chequeos (test-driven).
  - para comprobar que el software se comporta siempre y en todo momento como se espera que lo haga.
- ✓ Desarrollo 2: el trabajo se hace siempre en parejas
  - para compartir el conocimiento, la experiencia y las ideas.
- ✓ Desarrollo 3: todo el código desarrollado es de todos (propiedad colectiva del resultado)
  - para compartir la responsabilidad.
- ✓ Desarrollo 4: se integra continuamente (no dejar "fleclos sueltos" ni partes aisladas)
  - para reducir el impacto de añadir nuevas funcionalidades.
- ✓ Empresa 1: el cliente ha de ser parte integrante del equipo (y ha de estar accesible en cualquier momento)
  - para proporcionar, de forma directa y precisa, cuantas indicaciones se requieran sobre las especificaciones.

## CAPÍTULO 2: Investigación y Análisis de las Tecnologías Libres para el Desarrollo de Sistemas de Ayuda para SIG

---

- ✓ Empresa 2: jugar al juego de la planificación (tener un calendario y un plan sobre la mayor parte del trabajo)
  - para no ir a la deriva, sino manteniendo un rumbo y sabiendo nuestra situación. El destino final puede variar, pero en ningún momento se ha de estar perdidos sin saber a dónde se va.
- ✓ Empresa 3: ofrecer resultados regularmente (lo hecho hasta el momento, es funcional y ya resuelve algunas partes del proyecto –las más importantes–)
  - para dar al cliente retornos de la inversión que ha realizado hasta ese momento.
- ✓ Empresa 4: trabajar siempre a un ritmo sostenible (ajustar el proyecto, no las horas de trabajo diarias). **(Calero, 2003)**

### **Sobre Ciclos de Trabajo**

En cada ciclo de iteración (que tiene un tiempo fijo), el cliente tan solo puede cambiar el alcance y/o las prioridades del equipo de programadores. Durante una iteración, los programadores trabajan sobre las tareas que se han fijado antes de comenzar la iteración. Tras cada iteración, en el análisis y discusión de ésta, es cuando se decide el trabajo a desarrollar en la siguiente iteración. También es cuando se realizan los ajustes pertinentes en el plan general, el presupuesto y/o el calendario del proyecto.

**(Calero, 2003)**

### Qué tareas realizar en cada iteración y en qué orden hacerlo:

- ✓ Se decide en función de la importancia de las tareas para el cliente y en función del riesgo asociado a cada una de ellas. Las partes con mas “beneficio” (importancia para el cliente) y con menos “riesgo” (incertidumbre en su implementación) son las candidatas a ser abordadas antes, siempre se intenta maximizar el resultado obtenido en cada iteración.
- ✓ Quien lo decide es el cliente. Según él lo estime oportuno.
- ✓ El equipo de programadores tan solo asesora al cliente. Estimando el grado de complejidad y el tiempo que va a llevar cada parte del proyecto. (Los resultados de cada iteración son de gran ayuda para afinar la precisión de dichas estimaciones, como “Team Velocity”, capacidad de trabajo real del equipo.)

### Cómo realizar las tareas y qué recursos emplear en cada una:

- ✓ Lo decide el equipo de programadores. Ellos son quienes conocen la forma de realizar el trabajo.



## CAPÍTULO 2: Investigación y Análisis de las Tecnologías Libres para el Desarrollo de Sistemas de Ayuda para SIG

---

- ✓ El cliente solo aclara dudas que surjan sobre las especificaciones. Y fija la importancia que tiene cada elemento del trabajo, (las prioridades). **(Calero, 2003)**

Se pudiera afirmar que otras metodologías robustas como RUP y MSF están diseñadas para proyectos de largo alcance y a largo plazo, a diferencia de XP, destinada para organizar métodos de programación rápida y emergente.

Para el desarrollo de la solución se seleccionó XP, por posibilitar la implementación ágil de aplicaciones donde las decisiones y necesidades de implementación las determina el equipo de desarrollo, permitiendo el lanzamiento de versiones en periodos de tres a cuatro meses y la reorganización del código a medida que vayan surgiendo nuevos requerimientos sin tener en cuenta el surgimiento de nuevos riesgos.

### **2.6. Javascript Extendido (Ext JS).**

Ext JS es una API<sup>2</sup> escrito en Java Script con la finalidad de asistir el desarrollo de aplicaciones enriquecidas para internet (RIA, por sus siglas en inglés Rich Internet Applications). Entre las principales características de Ext JS resalta un fuerte paradigma basado en componentes soportado por recursos para la programación orientada a objetos en Java Script que facilitan la implementación de extensiones y aplicaciones de gran complejidad, el poseer un amplio conjunto de widgets configurables de alta calidad y una implementación transparente y sencilla para el trabajo con AJAX.**(Lobo, 2006)**

Quizás uno de los mayores éxitos de este API esté en la amplia, detallada y bien elaborada documentación, en la cantidad de ejemplos y en la poderosa comunidad detrás de este desarrollo. Ext JS fue creado inicialmente por Jack Slocum como extensión para Yahoo User Interface (YUI) otra librería desarrollada en Java Script que explota las potencialidades de AJAX para el desarrollo de RIA.

De manera muy sucinta se puede afirmar que Ext JS posee una licencia dual, si usted va a desarrollar aplicaciones de software libre acorde a la Licencia Pública General (GPL, por sus siglas en inglés General Public Licence) entonces debe acogerse a los términos de GPL 3.0, en tanto si va a desarrollar una aplicación propietaria tendrá que aceptar la Licencia Comercial. **(Lobo, 2006)**

---

<sup>2</sup> Interfaz de Programación de Aplicaciones.

## CAPÍTULO 2: Investigación y Análisis de las Tecnologías Libres para el Desarrollo de Sistemas de Ayuda para SIG

---

Lo mejor de todo es que Ext JS ofrece innumerables ejemplos de las principales funcionalidades que debe poseer cualquier sitio o aplicación Web permitiendo así el desarrollo a gran escala de soluciones de este tipo, de manera ágil y fácil, además de contar con una enciclopedia de widgets<sup>3</sup> que hacen la interfaz amigable y agradable a la vista. Lamentablemente no abunda mucho la documentación en español, inconveniente que limita su asimilación hasta cierto punto al dificultar su aprendizaje.

Se decidió su uso por sus amplias facilidades de uso, contando con una rica fuente de ejemplos ya pre desarrollados, posibilitando la reutilización racional de código y el desarrollo vertiginoso de aplicaciones Web, además de garantizar la compatibilidad con los SIG a desarrollar en el centro de desarrollo Geysed para el que está diseñada la aplicación.

### **2.7. Entorno de Desarrollo Integrado (IDE).**

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Puede dedicarse en exclusiva a un solo lenguaje de programación o puede utilizarse para varios.

#### ***Netbeans***

Netbeans es un entorno de desarrollo, hecho principalmente para el lenguaje de programación Java, con una gran gama de plugins a disposición del desarrollador. Es un producto libre y gratuito sin restricciones de uso. Es un proyecto de código abierto de gran éxito con una amplia comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Micro Systems fundó el proyecto de código abierto Netbeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos. Desde julio de 2006, Netbeans es licenciado bajo la Common Development and Distribution License (CDDL), una licencia basada en la Mozilla Public License (MPL). **(Cerde, 2009)**

#### Características de Netbeans

- ✓ Permite el desarrollo de aplicaciones Web, escritorio, para móviles y empresas.
- ✓ Permite el uso de lenguajes como Java, C/C++, Ruby on Rails, PHP, Groovy, Python entre otros.
- ✓ Es completamente Multiplataforma.

---

<sup>3</sup>Widget: es una pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños que son ejecutados por un motor de *widgets*. Entre sus objetivos están dar fácil acceso a funciones frecuentemente usadas y proveer de información visual o como interfaz gráfica.

## CAPÍTULO 2: Investigación y Análisis de las Tecnologías Libres para el Desarrollo de Sistemas de Ayuda para SIG

---

- ✓ Instalación y actualización muy simple.
- ✓ Características visuales para el desarrollo Web.
- ✓ Creador gráfico de juegos para celulares.
- ✓ Utilidades para Arquitectura Orientada a Servicios (En inglés Service - oriented architecture) SOA y UML.
- ✓ Amplio soporte para PHP.

Se utiliza Netbeans por su carácter de software libre. En temas de utilidad éste posee innumerables plugins para infinidad de situaciones y necesidades que lo soportan, por otra parte tiene garantías a la hora de editar el HTML en correspondencia con la selección de lenguaje propuesto, combina y se adecua muy provechosamente al uso y reutilización de las librerías y las Apis de EXT JS. Además de tener en cuenta todas las características antes mencionadas, por cuestiones de rendimiento y en la demanda de recursos de la máquina Netbeans es mucho más ligero.

### **2.8. Framework**

Un framework es una estructura software compuesta por componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Los objetivos principales que persigue éste son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de programación como el uso de patrones.

Un framework Web, es como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web.

#### **CodeIgniter.**

CodeIgniter es un conjunto de herramientas para personas que construyen su aplicación Web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría si lo escribiese desde cero, suministrándole un rico juego de librerías para tareas comúnmente necesarias, así como una interfaz simple y estructura lógica para acceder a esas librerías. CodeIgniter le permite creativamente enfocarse en su proyecto minimizando la cantidad de código necesaria para una tarea dada. Éste se encuentra bajo una licencia Open Source Apache/BSD-style y usa el acercamiento

## CAPÍTULO 2: Investigación y Análisis de las Tecnologías Libres para el Desarrollo de Sistemas de Ayuda para SIG

---

Modelo-Vista-Controlador, que permite una buena separación entre lógica y presentación. Esto es particularmente bueno para proyectos en los cuales diseñadores están trabajando con sus archivos de plantilla, ya que el código en esos archivos es mínimo. **(Lozano, 2006)**

### Características de CodeIgniter:

- ✓ Sistema basado en Modelo-Vista-Controlador.
- ✓ Compatible con PHP 4.
- ✓ Soporte de Active Record para Base de Datos.
- ✓ Formulario y Validación de Datos.
- ✓ Clase de Envío de Email. Soporta Archivos Adjuntos, email de texto/HTML, múltiples protocolos (sendmail, SMTP, and Mail) y más.
- ✓ Clase de FTP<sup>4</sup>.
- ✓ Localización.
- ✓ Paginación.
- ✓ Encriptación de Datos.
- ✓ URLs amigables a motores de búsqueda.

### **2.9. Conclusiones.**

Se puede afirmar que el conjunto de tecnologías y herramientas seleccionadas para el desarrollo de la solución propuesta son utilizadas en la universidad y en el centro de desarrollo GEYSED por sus desarrolladores, garantizando la compatibilidad con los SIG en desarrollo y el soporte de las mismas. Partiendo de un análisis de éstas se seleccionó XP como metodología de desarrollo más eficiente para la implementación de la aplicación y finalmente la selección de NetBeans, IDE sobre el cual con la integración de PHP y EXT JS como lenguajes de programación Web y librería de Javascript respectivamente, sobre el framework seleccionado, se obtendrá un producto soberano y de calidad.

---

<sup>4</sup> FTP: File Transfer Protocol (Protocolo de Transferencia de Archivos).

## CAPÍTULO 3: Presentación de la Solución Propuesta.

### 3.1. *Introducción.*

Para lograr la implementación satisfactoria de la aplicación, es necesario el estudio y puesta en práctica de XP a través del ciclo de vida propuesto por la misma como metodología de desarrollo ágil, ésta cuenta con 4 fases, Exploración, Planificación, Implantación y Pruebas. En este capítulo se desglosan y describen los requisitos funcionales de la aplicación en Historias de Usuario (HU) descritas exactamente como fueron concebidas por los usuarios de la solución propuesta y toda la planificación del Proyecto (Exploración y Planificación).

### 3.2. *Propuesta del Sistema*

Se propone el desarrollo de un sistema que permita la creación de Sistemas de Ayudas personalizados específicamente para Sistemas de Información Geográfica de forma dinámica y ágil, y que sea a su vez desarrollado con tecnologías libres, en vistas a estandarizar y viabilizar el proceso de desarrollo de SIG en el Centro GEYSED de la Facultad 6 de la UCI.

### 3.3. *Personas Relacionadas con el Sistema*

<b>Personas relacionadas con el sistema</b>	<b>Justificación</b>
Usuario o Desarrollador	Es la persona que interactúa con el sistema encargado de gestionar las Ayudas.

Tabla # 1 Personas Relacionadas con el Sistema

### 3.4. *Fase de Exploración*

La primera fase de la metodología de desarrollo XP es Exploración, en esta fase los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Así mismo el equipo de trabajo se familiariza con la tecnología, las herramientas y prácticas a utilizar en el proyecto. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología. Finalmente se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. Es la fase en la que se define el alcance general del proyecto. **(Letelier y Penadés, 2003)**

### **3.4.1. Requisitos no Funcionales**

Los requisitos no funcionales son propiedades o cualidades que el producto debe cumplir. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

#### **Interfaz y Usabilidad:**

La interfaz debe ser sencilla, clara, intuitiva y amigable, manteniendo un formato estándar en todas las páginas. No cuenta con gran cantidad de imágenes, garantizando que el tiempo de descarga sea mínimo y así no demorar las respuestas al usuario. Teniendo en cuenta el fin para el que se desarrolla, la aplicación debe ser seria, formal y tener una navegación sugerente independientemente de que los usuarios o clientes sean desarrolladores de software con preparación para su manejo.

#### **Rendimiento:**

La capacidad de respuesta debe ser bien independiente de la plataforma y recursos del ordenador, teniendo como prioridad garantizar que el tiempo de respuesta a las solicitudes de los usuarios al sistema sea mínimo, o sea, en un período de tiempo breve (escasos segundos). El sistema debe ser lo más estable, ligero y confiable posible.

#### **Portabilidad:**

El sistema debe ser multiplataforma, éste debe ajustarse totalmente con la mayor brevedad, independientemente de que las herramientas para su desarrollo sean preferentemente software libre.

#### **Seguridad:**

La información manejada por el sistema cuenta de protección ante intrusos y accesos no autorizados, es vista únicamente por aquellos usuarios que tienen derecho a verla. El sistema debe priorizar la identificación del usuario antes de que sea capaz de realizar cualquier acción sobre el sistema.

#### **Hardware:**

Requisitos mínimos: Procesador Pentium 3 a 2.0 GHz, 256 de RAM, disponibilidad en disco duro mínima de: 100 MB. 256 MB de Video OnBoard.

#### **Software.**

Sistema Operativo: Ubuntu, o Windows SP2 en adelante, Navegador WEB, Servidor WEB y Gestor de Base de Datos PostgreSQL.

### **3.4.2. Historias de Usuario**

Las Historias de Usuario (HU) son la técnica utilizada para especificar los requisitos del software. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el

## CAPÍTULO 3: Presentación de la Solución Propuesta

---

sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las HU es muy dinámico y flexible. Cada HU es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. **(Canós, 2010)**

### Características de las HU

- ✓ Las escriben los propios clientes, tal y como ven ellos las necesidades del sistema.
- ✓ Las Historias de Usuario son similares al empleo de escenarios, con la excepción de que no se limitan a la descripción de la interfaz de usuario.
- ✓ También conducirán el proceso de creación de los test de aceptación (empleados para verificar que las Historias de Usuario han sido implementadas correctamente).
- ✓ Existen diferencias entre éstas y la tradicional especificación de requisitos. La principal diferencia es el nivel de detalle. Las Historias de Usuario solamente proporcionaran los detalles sobre la estimación del riesgo y cuánto tiempo conlleva la implementación de la misma.
- ✓ Las Historias de Usuario son descompuestas en tareas de programación y asignadas a los programadores para ser implementadas durante una iteración.

Las Historias de Usuarios son muy similares a los Casos de Uso convencionales y tienen los mismos objetivos **(Beck, 2005)**

<b>Historia de Usuario</b>	
<b>Número:</b> Número sucesivo a partir de 1.	<b>Nombre de Historia:</b> Identifica la historia de usuario en cuestión.
<b>Usuario:</b> Usuarios que interactúan con la HU.	
<b>Prioridad en el Negocio:</b> Define la relevancia e impacto de la Historia de Usuario para el negocio de acuerdo con las necesidades del usuario (Alta / Media / Baja).	<b>Riesgo en Desarrollo:</b> Define la dificultad técnica desde el punto de vista del programador (Alto / Medio / Bajo).
<b>Puntos de Estimación:</b> La estimación de esfuerzo se expresa utilizando como medida el punto. Un punto se	<b>Iteración Asignada:</b> Precisa la iteración a la que pertenece la Historia de

## CAPÍTULO 3: Presentación de la Solución Propuesta

---

considera como una semana ideal de trabajo.	Usuario.
<b>Descripción:</b> Explica en qué consiste la Historia de Usuario, teniendo en cuenta las acciones realizadas por el usuario y la respuesta brindada por el sistema.	
<b>Observaciones:</b> Información extra que se estime agregar para hacer más comprensible la Historia de Usuario.	

Tabla # 2 Historia de Usuario.

### 3.4.3. Historias de Usuario Definidas

<b>Historia de Usuario</b>	
<b>Número:</b> 1.	<b>Nombre de Historia:</b> Crear Proyecto.
<b>Usuario:</b> Cliente o Desarrollador.	
<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo en Desarrollo:</b> Medio.
<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 1.
<b>Descripción:</b> El cliente inserta el nombre del proyecto en cuestión y el sistema inserta el mismo en base de datos y visualiza el proyecto creado.	
<b>Observaciones:</b> El sistema guarda el nombre del proyecto creado en base de datos y crea un árbol jerárquico donde el nombre del proyecto ocupa el nodo raíz del mismo.	

Tabla # 3 HU Crear Proyecto.

<b>Historia de Usuario</b>	
<b>Número:</b> 2.	<b>Nombre de Historia:</b> Editar Proyecto.
<b>Usuario:</b> Cliente o Desarrollador.	
<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo en Desarrollo:</b> Medio.



## CAPÍTULO 3: Presentación de la Solución Propuesta

---

<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 1.
<b>Descripción:</b> El cliente selecciona un proyecto X, y el sistema proporciona el acceso al mismo para su edición guardando los cambios realizados por el usuario y lo visualiza.	
<b>Observaciones:</b> El cliente selecciona el proyecto antes creado y el sistema brinda la posibilidad de hacer cualquier cambio en el mismo y lo visualiza.	

Tabla # 4 HU Editar Proyecto.

<b>Historia de Usuario</b>	
<b>Número:</b> 3.	<b>Nombre de Historia:</b> Eliminar Proyecto.
<b>Usuario:</b> Cliente o Desarrollador.	
<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo en Desarrollo:</b> Medio.
<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 1.
<b>Descripción:</b> El cliente selecciona un proyecto X, y el sistema proporciona el acceso al mismo para ser eliminado y lo visualiza.	
<b>Observaciones:</b> El cliente selecciona el proyecto antes creado y el sistema brinda la posibilidad de eliminar el mismo y todos los nodos hijos que lo conforman.	

Tabla # 5 HU Eliminar Proyecto.

Se confeccionan las HU Adicionar Categoría, Editar Categoría y Eliminar Categoría. (Ver Anexo # 1)

<b>Historia de Usuario</b>	
<b>Número:</b> 7.	<b>Nombre de Historia:</b> Adicionar Ficha.
<b>Usuario:</b> Cliente o Desarrollador.	
<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo en Desarrollo:</b> Medio.

## CAPÍTULO 3: Presentación de la Solución Propuesta

---

<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 3.
<b>Descripción:</b> El cliente selecciona una categoría (Y), y el sistema brinda la posibilidad de adicionarle una ficha (Z) y lo visualiza.	
<b>Observaciones:</b> El cliente selecciona el proyecto (X) antes creado y el sistema brinda la posibilidad de seleccionar una de sus (Y) categorías y a su vez adicionarle una ficha (Z) formando un árbol jerárquico donde (X) proyecto es nodo raíz, (Y) categorías sub nodo de (X) proyecto, y a su vez (Z) ficha es subnodo de (Y) categoría.	

Tabla # 6 Adicionar Ficha.

<b>Historia de Usuario</b>	
<b>Número:</b> 8.	<b>Nombre de Historia:</b> Editar Ficha.
<b>Usuario:</b> Cliente o Desarrollador.	
<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo en Desarrollo:</b> Medio.
<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 3.
<b>Descripción:</b> El cliente selecciona una ficha (Z), y el sistema brinda la posibilidad de editarla, guardar sus cambios y lo visualiza.	
<b>Observaciones:</b> El cliente selecciona el proyecto (X) antes creado y el sistema brinda la posibilidad de seleccionar una de sus (Y) categorías y a su vez seleccionar una ficha (Z), editarla, guardar los cambios y visualizarlos.	

Tabla # 7 HU Editar Ficha.

<b>Historia de Usuario</b>	
<b>Número:</b> 9.	<b>Nombre de Historia:</b> Eliminar Ficha.
<b>Usuario:</b> Cliente o Desarrollador.	

## CAPÍTULO 3: Presentación de la Solución Propuesta

---

<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo en Desarrollo:</b> Medio.
<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 3.
<b>Descripción:</b> El cliente selecciona una ficha (Z), y el sistema brinda la posibilidad de eliminarla y lo visualiza.	
<b>Observaciones:</b> El cliente selecciona el proyecto (X) antes creado y el sistema brinda la posibilidad de seleccionar una de sus (Y) categorías y a su vez seleccionar una ficha (Z), eliminarla del árbol jerárquico que conforma y lo visualiza.	

Tabla # 8 HU Eliminar Ficha.

<b>Historia de Usuario</b>	
<b>Número:</b> 10.	<b>Nombre de Historia:</b> Adicionar Contenido.
<b>Usuario:</b> Cliente o Desarrollador.	
<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo en Desarrollo:</b> Medio.
<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 4.
Descripción: El cliente selecciona una ficha (Z), y el sistema brinda la posibilidad de adicionar y configurar su contenido.	
Observaciones: El cliente selecciona el proyecto (X) antes creado y el sistema brinda la posibilidad de seleccionar una de sus (Y) categorías y a su vez seleccionar una ficha (Z), adicionar y configurar su contenido.	

Tabla # 9 HU Adicionar Contenido.

<b>Historia de Usuario</b>	
<b>Número:</b> 11.	<b>Nombre de Historia:</b> Editar Contenido.
<b>Usuario:</b> Cliente o Desarrollador.	

## CAPÍTULO 3: Presentación de la Solución Propuesta

---

<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo en Desarrollo:</b> Medio.
<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 4.
<b>Descripción:</b> El cliente selecciona una ficha (Z) y el sistema brinda la posibilidad de seleccionar su contenido y editarlo.	
<b>Observaciones:</b> El cliente selecciona el proyecto (X) antes creado y el sistema brinda la posibilidad de seleccionar una de sus (Y) categorías y a su vez seleccionar una ficha (Z), y editar su contenido.	

Tabla # 10 HU Editar Contenido.

<b>Historia de Usuario</b>	
<b>Número:</b> 12.	<b>Nombre de Historia:</b> Eliminar Contenido.
<b>Usuario:</b> Cliente o Desarrollador.	
<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo en Desarrollo:</b> Medio.
<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 4.
<b>Descripción:</b> El cliente selecciona una ficha (Z) y el sistema brinda la posibilidad de seleccionar su contenido y eliminarlo.	
<b>Observaciones:</b> El cliente selecciona el proyecto (X) antes creado y el sistema brinda la posibilidad de seleccionar una de sus (Y) categorías y a su vez seleccionar una ficha (Z), y eliminar su contenido.	

Tabla # 11 Eliminar Contenido.

<b>Historia de Usuario</b>	
<b>Número:</b> 13.	<b>Nombre de Historia:</b> Previsualizar Proyecto.
<b>Usuario:</b> Cliente o Desarrollador.	
<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo en Desarrollo:</b> Alta.

## CAPÍTULO 3: Presentación de la Solución Propuesta

---

<b>Puntos de Estimación:</b> 3.	<b>Iteración Asignada:</b> 5.
<b>Descripción:</b> El cliente selecciona un proyecto antes creado y el sistema muestra como está conformada la aplicación resultante.	
<b>Observaciones:</b> El cliente selecciona el proyecto (X) antes creado y el sistema muestra la aplicación resultante con su árbol jerárquico completamente.	

Tabla # 12 Previsualizar Proyecto.

<b>Historia de Usuario</b>	
<b>Número:</b> 14.	<b>Nombre de Historia:</b> Exportar Proyecto.
<b>Usuario:</b> Cliente o Desarrollador.	
<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo en Desarrollo:</b> Alta.
<b>Puntos de Estimación:</b> 3.	<b>Iteración Asignada:</b> 6.
<b>Descripción:</b> El cliente selecciona un proyecto (X) antes creado y el sistema guarda la aplicación resultante en un fichero definido por el usuario o cliente.	
<b>Observaciones:</b> El cliente selecciona el proyecto (X) antes creado y el sistema guarda todo el sistema de archivos de la aplicación Web resultante en un fichero predeterminado.	

Tabla # 13 HU Importar Proyecto.

<b>Historia de Usuario</b>	
<b>Número:</b> 15	<b>Nombre de Historia:</b> Adicionar Glosario.
<b>Usuario:</b> Cliente o Desarrollador.	
<b>Prioridad en el Negocio:</b> Media.	<b>Riesgo en Desarrollo:</b> Media.
<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 7.

## CAPÍTULO 3: Presentación de la Solución Propuesta

<b>Descripción:</b> El cliente selecciona un proyecto (X) y adiciona el glosario, el sistema muestra la interfaz definida para esta acción permitiéndole al usuario o cliente agregar palabras y su significado.
<b>Observaciones:</b> El cliente selecciona el proyecto (X) antes creado, accede al botón Glosario y el sistema muestra una interfaz que le permite al usuario insertar palabras y el significado de la misma, finalmente el sistema muestra en la aplicación Web generada el glosario de términos configurado por el cliente.

Tabla # 14 HU Adicionar Glosario.

Historia de Usuario	
<b>Número:</b> 16	<b>Nombre de Historia:</b> Editar Glosario.
<b>Usuario:</b> Cliente o Desarrollador.	
<b>Prioridad en el Negocio:</b> Media.	<b>Riesgo en Desarrollo:</b> Media.
<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 7.
<b>Descripción:</b> El cliente selecciona un proyecto (X) y edita el glosario, el sistema muestra la interfaz definida para esta acción permitiéndole al usuario o cliente editar las palabras y el significado.	
<b>Observaciones:</b> El cliente selecciona el proyecto (X) antes creado, accede al botón Glosario y el sistema muestra una interfaz que le permite al usuario editar palabras y el significado de la misma, finalmente el sistema muestra en la aplicación Web generada el glosario de términos configurado por el cliente.	

Tabla # 15 HU Editar Glosario.

Historia de Usuario	
<b>Número:</b> 17	<b>Nombre de Historia:</b> Eliminar Glosario.
<b>Usuario:</b> Cliente o Desarrollador.	

## CAPÍTULO 3: Presentación de la Solución Propuesta

<b>Prioridad en el Negocio:</b> Media.	<b>Riesgo en Desarrollo:</b> Media.
<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 7.
<b>Descripción:</b> El cliente selecciona un proyecto (X) y eliminar el glosario, el sistema muestra la interfaz definida para esta acción permitiéndole al usuario o cliente eliminar el glosario.	
<b>Observaciones:</b> El cliente selecciona el proyecto (X) antes creado, accede al botón Glosario y el sistema muestra una interfaz que le permite al usuario eliminar el glosario.	

Tabla # 16 HU Eliminar Glosario.

<b>Historia de Usuario</b>	
<b>Número:</b> 18	<b>Nombre de Historia:</b> Autenticar.
<b>Usuario:</b> Cliente o Desarrollador	
<b>Prioridad en el Negocio:</b> Baja.	<b>Riesgo en Desarrollo:</b> Bajo.
<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 7.
<b>Descripción:</b> El usuario o cliente se autentica y accede al sistema.	
<b>Observaciones:</b> El usuario o cliente se autentica entrando usuario y contraseña.	

Tabla # 17 HU Autenticar.

### 3.5. Planificación

En la siguiente fase se hace un estudio y planificación del tiempo y el esfuerzo que conlleva el desarrollo de la aplicación propuesta, en dependencia de las tecnologías a usar y los conocimientos del equipo de desarrollo, así como la distribución de la cantidad de iteraciones y HU definidas en la fase de Exploración. Además de los planes de entrega propuestos partiendo de todo lo antes expuesto.

## CAPÍTULO 3: Presentación de la Solución Propuesta

---

### 3.5.1. Estimación de Esfuerzos

En la presentación de la solución propuesta se realizó la estimación de esfuerzos siguiente:

<b>Historias de Usuarios</b>	<b>Puntos de Estimación</b>
Crear Proyecto.	1
Editar Proyecto.	
Eliminar Proyecto.	
Crear Categoría.	1
Editar Categoría.	
Eliminar Categoría.	
Crear Ficha.	1
Editar Ficha.	
Eliminar Ficha.	
Crear Contenido.	1
Editar Contenido.	
Eliminar Contenido.	
Previsualizar Proyecto.	3
Exportar Proyecto.	3
Adicionar Glosario.	1
Editar Glosario.	
Eliminar Glosario.	



## CAPÍTULO 3: Presentación de la Solución Propuesta

---

Autenticar	
------------	--

Tabla # 18 Estimación de Esfuerzos.

### **3.5.2. Plan de Iteraciones.**

En esta fase se realizan varias iteraciones sobre la aplicación con el objetivo de refinar y solidificar los resultados finales en cada iteración, y que al final de la última se entregue un producto funcional y listo para su comercialización y despliegue. En cada una de ellas no debe transcurrir un tiempo mayor de 3 semanas. Una vez definidas las HU y estimado el esfuerzo para cada una de éstas se tomó la decisión de desarrollar la aplicación en 7 iteraciones.

**Iteración # 1:** Se implementan las Historias de Usuario; “Crear Proyecto”, “Editar Proyecto”, y “Eliminar Proyecto” de alta prioridad. Al finalizar la iteración el cliente gestiona los proyectos y tiene una idea general de cómo queda y funciona la gestión de los demás componentes, pues son muy similares en cuanto a complejidad y funcionalidad, además de obtener un prototipo de Interfaz de Usuario cuestionable.

**Iteración # 2:** Se implementan las Historias de Usuario; “Adicionar Categoría”, “Editar Categoría”, y “Eliminar Categoría” de alta prioridad, al final de esta iteración se gestiona las categorías y se tiene la primera idea lógica del funcionamiento jerárquico de los componentes que conforman la aplicación de ayuda.

**Iteración # 3:** Se implementan las Historias de Usuario; “Adicionar Ficha”, “Editar Ficha”, y “Eliminar Ficha” de alta prioridad, permitiéndole al usuario la gestión de las fichas.

**Iteración # 4:** Se implementan las Historias de Usuario; “Adicionar Contenido”, “Editar Contenido”, y “Eliminar Contenido” de alta prioridad, permitiéndole al usuario la gestión de los contenidos.

**Iteración # 5:** Se implementa la Historia de Usuario; “Previsualizar Proyecto” de alta prioridad, donde el usuario tiene la posibilidad de ver de forma dinámica cómo queda su aplicación Web antes de importar los archivos conformados por la aplicación.

**Iteración # 6:** Se implementa la Historia de Usuario; “Exportar Proyecto” de alta prioridad, donde el usuario finalmente conforma la aplicación Web resultante y guarda en un fichero predeterminado el producto generado por la aplicación.

**Iteración # 7:** Se implementan las Historias de Usuario; “Adicionar Glosario”, “Editar Glosario”, “Eliminar Glosario” y “Autenticar” de mediana prioridad, permitiéndole al usuario o cliente el acceso a la

## CAPÍTULO 3: Presentación de la Solución Propuesta

aplicación y la gestión del glosario de términos en vistas a aumentar el entendimiento y usabilidad de la aplicación Web generada.

### 3.5.3. Plan de duración de las Iteraciones

Con el objetivo de garantizar la alta resolutivez del proceso de desarrollo se confecciona un plan de iteraciones. El plan de iteraciones contiene de manera organizada el tiempo de duración de cada iteración así como el orden en que se van a implementar las Historias de Usuario definidas con anterioridad.

Partiendo de las HU definidas y la descripción de cada iteración se definió el siguiente *Plan de Duración de las Iteraciones*.

Iteración	Historias de Usuario	Duración Total de la Iteración
1	Crear Proyecto.	1 Semana.
	Editar Proyecto.	
	Eliminar Proyecto.	
2	Adicionar Categoría.	1 Semana.
	Editar Categoría.	
	Eliminar Categoría.	
3	Adicionar Ficha.	1 Semana.
	Editar Ficha.	
	Eliminar Ficha.	
4	Adicionar Contenido.	1 Semana.
	Editar Contenido.	
	Eliminar Contenido.	
5	Previsualizar Proyecto.	3 Semanas.
6	Exportar Proyecto.	3 Semanas.
7	Adicionar Glosario.	1 Semana.
	Editar Glosario.	
	Eliminar Glosario.	
	Autenticar	

Tabla # 19 Plan de Duración de las Iteraciones.

## CAPÍTULO 3: Presentación de la Solución Propuesta

---

### 3.5.4. Plan de Entrega

A continuación se muestra el Plan de Entrega confeccionado partiendo de los puntos de estimación definidos en las Historias de Usuario y el plan de duración de las iteraciones antes ilustrado.

Número de la Iteración	Duración de la Iteración	Fecha de Inicio	Fecha Final
1	1	01/04/2012	07/04/2012
2	1	08/04/2012	15/04/2012
3	1	16/04/2012	23/04/2012
4	1	24/04/2012	30/04/2012
5	3	01/05/2012	22/05/2012
6	3	23/05/2012	15/06/2012
7	1	16/06/2012	23/06/2012

Tabla # 20 Plan de Entrega.

### 3.6. Conclusiones.

En este capítulo se desarrolló la descripción de la solución propuesta para la implementación de una plataforma de gestión de sistemas de ayuda para Sistemas de Información Geográfica en entornos Web, basada en las dos primeras fases del ciclo de vida de XP como metodología de desarrollo ágil (Exploración y Planificación), donde la primera de éstas permitió la descripción del funcionamiento de la aplicación y conocer los deseos del cliente y su visión acerca de lo que necesitaba. Y la segunda dio la posibilidad de estimar el esfuerzo necesario y planificar el tiempo de duración del proyecto partiendo de la distribución de las tareas organizadas en iteraciones. Finalmente se llegó a la conclusión de que el desarrollo del proyecto tendrá una duración aproximada de 11 semanas según el plan de entrega confeccionado al final de este capítulo.

### CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta.

#### 4.1. *Introducción*

El diseño y desarrollo es la parte más importante en el proceso de la programación extrema, pues éstos van conjuntamente de la mano. Todos los trabajos tienen como objetivo que se programen lo más rápidamente posible, sin interrupciones y en dirección correcta, partiendo del diseño preconcebido por el equipo de desarrollo soportado todo el tiempo por la opinión del cliente como pilar fundamental del proceso XP. Además se establecen los mecanismos para que éste sea revisado y mejorado de manera continuada a lo largo del proyecto según se van añadiendo funcionalidades al mismo, de esta manera cada uno de los fragmentos de códigos escritos en cada una de las iteraciones son probados y evaluados con el objetivo de ganar en eficiencia y confianza a la hora de entregar un producto.

A continuación se hace alusión a las últimas dos fases del ciclo de vida de XP donde se diseñan y escriben Tarjetas C.R.C<sup>5</sup>, se presentan las iteraciones y las tareas de ingeniería generadas por cada una de las Historias de Usuario y los test o pruebas realizados durante la construcción y terminación de la aplicación propuesta.

#### 4.2. *Diseño*

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga cuesta menos tiempo y esfuerzo desarrollar. El uso de términos sugerentes y una correcta especificación de los nombres de métodos y clases ayudan a comprender el diseño y facilita la posterior ampliación y la reutilización del código.

El uso de las tarjetas C.R.C., permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica. Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad. **(Beck, 2005)**

---

<sup>5</sup> Tarjetas C.R.C.: Tarjetas de las Clases, Responsabilidades y Colaboración

## 4.2.1. Tarjetas CRC

Tarjeta CRC	
<b>Clase:</b> GestionarAyudaController	
<b>Súper Clase:</b> -	
<b>Sub Clase:</b> -	
<b>Responsabilidades:</b> <ul style="list-style-type: none"><li>• getProyectos()</li><li>• adicionarProyecto()</li><li>• editarProyecto()</li><li>• eliminarProyecto()</li><li>• getCategorias()</li><li>• adicionarCategoria()</li><li>• editarCategoria()</li><li>• eliminarCategoria()</li><li>• getFichas()</li><li>• adicionarFicha()</li><li>• editarFicha()</li><li>• eliminarFicha()</li><li>• getContenidos()</li><li>• adicionarContenido()</li><li>• editarContenido()</li><li>• eliminarContenido()</li></ul>	<b>Colaboraciones:</b> <ul style="list-style-type: none"><li>• CI_Controller</li><li>• claseArbol</li><li>• GestionarAyudaModel</li></ul>

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

<ul style="list-style-type: none"><li>• getGlosarios()</li><li>• getGlosarioProyecto()</li><li>• adicionarGlosario()</li><li>• editarGlosario()</li><li>• eliminarGlosario()</li></ul>	
--	--

Tabla # 21 Tarjeta CRC GestionarAyudaController

Tarjeta CRC	
<b>Clase:</b> GestionarAyudaModel	
<b>Súper Clase:</b> -	
<b>Sub Clase:</b> -	
<b>Responsabilidades:</b> <ul style="list-style-type: none"><li>• Accede a la estructura de datos, interviene en la recuperación, inserción y actualización de la información en la base de datos.</li></ul>	<b>Colaboraciones:</b> <ul style="list-style-type: none"><li>• CI_Model</li><li>• ProyectoDao</li><li>• ContenidoDao</li><li>• FichaDao</li><li>• GlosarioDao</li><li>• CategoriaDao</li></ul>

Tabla # 22 Tarjeta CRC GestionarAyudaModel

### 4.2.2. Arquitectura del Sistema

Una arquitectura de software, también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. La arquitectura de software establece los fundamentos para que analistas, diseñadores, programadores y demás miembros de equipo trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades. Además define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. **(Morales, 2008)**

La arquitectura de software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos, el ambiente, los principios que orientan su diseño y evolución.

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

---

La Arquitectura de Software es una vista abstracta del funcionamiento de un sistema software, es la descripción de la colaboración que existe entre las partes o componentes de una aplicación, la misma puede ser vista o modelada de disímiles maneras ya que la misma depende en gran medida de aspectos como rendimiento, funcionalidad, y flexibilidad disponibles, así como los conocimientos del arquitecto que la diseñe.

Para el desarrollo Web, la solución más común para la organización de su código de hoy en día es el patrón de arquitectura Modelo Vista Controlador (MVC). En resumen, el patrón de arquitectura MVC define una manera de organizar el código de acuerdo a su naturaleza. Este patrón separa el código en tres capas:

- ✓ La capa Modelo define la lógica de negocio (la base de datos pertenece a esta capa).
- ✓ La Vista es con lo que el usuario interactúa (un motor de plantillas es parte de esta capa).
- ✓ El Controlador es la pieza de código que llama al Modelo para obtener algunos datos que le pasa a la Vista para la presentación al cliente.

### **4.2.2.1. Patrones de Diseño en Codelgniter.**

Desde un punto de vista técnico y arquitectónico, Codelgniter fue creado con los siguientes objetivos:

- ✓ Instanciación Dinámica: Los componentes son cargados y las rutinas ejecutadas sólo cuando se pide, en vez de globalmente. No se hacen suposiciones por el sistema acerca de qué puede necesitar además de los recursos mínimos del núcleo, así que el sistema es muy liviano por defecto. Los eventos, que se desencadenan por la petición HTTP, y los controladores y vistas que diseñe determinan que es invocado.
- ✓ Poco Acoplamiento: Acoplamiento es el grado que los componentes de un sistema dependen entre ellos. Mientras menos componentes dependan de otro, más reusable y flexible el sistema se convierte.
- ✓ Singularidad del Componente: Mientras más estrecho es el enfoque de los componentes, mayor es el grado de singularidad de los mismos. En Codelgniter, cada clase y sus funciones son altamente autónomas para permitir máxima utilidad.

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

---

CodeIgniter es un sistema instanciado dinámicamente, poco acoplado con gran singularidad de componente. Garantizando la simplicidad, flexibilidad y alto performance en un paquete de pequeña pisada. (Lozano, 2006)

### 4.2.2.2. *Patrones de Diseño.*

Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Son soluciones basadas en la experiencia de las comunidades de desarrollo.

Un patrón de diseño es:

- ✓ Una solución estándar para un problema común de programación.
- ✓ Una técnica para flexibilizar el código haciéndolo satisfacer ciertos criterios.
- ✓ Un proyecto o estructura de implementación que logra una finalidad determinada.
- ✓ Un lenguaje de programación de alto nivel.
- ✓ Una manera más práctica de describir ciertos aspectos de la organización de un programa.
- ✓ Conexiones entre componentes de programas.
- ✓ La forma de un diagrama de objeto o de un modelo de objeto.

Para la implementación de la solución propuesta se usaron los diferentes patrones de diseño:

- ✓ Bajo Acoplamiento: Debe haber pocas dependencias entre las clases, uno de los principales síntomas de un mal diseño y alto acoplamiento es una herencia muy profunda.
- ✓ Fachada: Proporciona una interfaz unificada de alto nivel que, representando a todo un subsistema, facilite su uso. La “fachada” satisface a la mayoría de los clientes, sin ocultar las funciones de menor nivel a aquellos que necesiten acceder a ellas.
- ✓ Observador: Cuando un cambio en un objeto requiere cambiar otros objetos y no sabemos cuántos exactamente lo necesitan, o la manera de notificar sus cambios a otros objetos, sin necesidad de saber nada acerca de su identidad (acoplamiento débil).
- ✓ Iterador: Proporciona una forma de acceder secuencialmente a los elementos sin revelar su representación interna, itera sobre los elementos de una lista sin ampliar la interfaz, permitiendo recorridos simultáneos, garantizando múltiples recorridos sobre los objetos agregados.



## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

---

- ✓ **Creador:** El problema que trata este patrón es el de establecer quién es el responsable de crear una nueva instancia de una clase, de modo que este patrón asigna responsabilidades relacionadas con la creación de objetos, definiendo como objeto creador aquel que necesite conectarse al objeto creado, éste proporciona un diseño que puede soportar un bajo acoplamiento y una mayor claridad. **(Canales, 2004)**
- ✓ **MVC:** El patrón de arquitectura MVC define una manera de organizar el código de acuerdo a su naturaleza. Este patrón separa el código en tres capas, Modelo, Vista y Controlador.

### 4.3. Implementación.

#### DESARROLLO DE LAS ITERACIONES:

Durante la fase de Planificación se detallaron las Historias de Usuario correspondientes a cada una de las iteraciones a desarrollar, las cuales tienen como objetivo principal atender las necesidades requeridas por el cliente. En el transcurso de las iteraciones se lleva a cabo una revisión del plan de iteraciones, modificándose en caso de ser preciso. Como parte de este plan, se descomponen las Historias de Usuario en tareas de programación o ingeniería, asignando a un equipo de desarrollo (o una persona), responsable de su implementación, aplicando la práctica de la programación en parejas. Estas tareas no tienen que necesariamente ser entendidas por el cliente, pueden ser escritas en lenguaje técnico y son para el uso estricto de los programadores.

Teniendo en cuenta la planificación realizada con anterioridad, se llevó a cabo el desarrollo del sistema en siete iteraciones, obteniéndose como finalidad un producto con todas las restricciones y características deseadas por el cliente. A continuación se detallan cada una de las iteraciones.

#### 4.3.1. Iteración # 1.

Historias de Usuarios	Tiempo de Implementación	
	Estimación	Real
Crear Proyecto.	1	1
Editar Proyecto.		
Eliminar Proyecto.		

Tabla # 23 HUs Iteración 1.

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

---

Tarea de la Ingeniería	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 1.
<b>Nombre de la Tarea:</b> Crear la Interfaz Gráfica mediante la cual se gestionan y adicionan los proyectos.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 01/04/2012.	<b>Fecha Fin:</b> 03/04/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b> En esta Tarea se crea el grid necesario sobre el cual se lista la gestión de los proyectos.	

Tabla # 24 Tarea de Ingeniería # 1, HU # 1, Iteración # 1.

Tarea de la Ingeniería	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 2.
<b>Nombre de la Tarea:</b> Editar Proyecto y Visualizar Cambios.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.3.
<b>Fecha Inicio:</b> 04/04/2012.	<b>Fecha Fin:</b> 05/04/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b> Se edita el proyecto y se lista en el grid.	

Tabla # 25 Tarea de Ingeniería # 1, HU # 2, Iteración # 1.

Tarea de la Ingeniería	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 3.
<b>Nombre de la Tarea:</b> Eliminar Proyecto y Visualizar.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.3.
<b>Fecha Inicio:</b> 06/04/2012.	<b>Fecha Fin:</b> 07/04/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b> Se elimina el proyecto y se lista en el grid para visualizar los cambios.	

Tabla # 26 Tarea de Ingeniería # 1, HU # 3, Iteración # 1.

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

---

Se confeccionan las Tareas de Ingeniería que conforman la Iteración # 2. (Ver Anexo # 2)

### 4.3.2. Iteración # 3.

Tarea de la Ingeniería		
Historias de Usuarios	Tiempo de Implementación	
	Estimación	Real
Adicionar Ficha.	1	1
Editar Ficha.		
Eliminar Ficha.		

Tabla # 27 HUs Iteración # 3.

Tarea de la Ingeniería	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 7.
<b>Nombre de la Tarea:</b> Crear la Interfaz Gráfica mediante la cual se gestionan las fichas y adicionan las mismas.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.3.
<b>Fecha Inicio:</b> 16/04/2012.	<b>Fecha Fin:</b> 19/04/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b> En esta Tarea se crea el grid necesario sobre el cual se lista la gestión de las Fichas.	

Tabla # 28 Tarea de Ingeniería # 1, HU # 7, Iteración # 3.

Tarea de la Ingeniería	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 8.
<b>Nombre de la Tarea:</b> Editar Ficha y Visualizar Cambios.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.3.
<b>Fecha Inicio:</b> 20/04/2012.	<b>Fecha Fin:</b> 21/04/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b> Se edita la Ficha y se lista en el grid.	

Tabla # 29 Tarea de Ingeniería # 1, HU # 8, Iteración # 3.

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

<b>Tarea de la Ingeniería</b>	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 9.
<b>Nombre de la Tarea:</b> Eliminar Ficha y Visualizar.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.3.
<b>Fecha Inicio:</b> 22/04/2012.	<b>Fecha Fin:</b> 23/04/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b> Se elimina la Ficha y se lista en el grid.	

Tabla # 30 Tarea de Ingeniería # 1, HU # 9, Iteración # 3.

### 4.3.3. Iteración # 4.

<b>Historias de Usuarios</b>	<b>Tiempo de Implementación</b>	
	<b>Estimación</b>	<b>Real</b>
Adicionar Contenido.	1	1
Editar Contenido.		
Eliminar Contenido.		

Tabla # 31 HUs Iteración # 4.

<b>Tarea de la Ingeniería</b>	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 10.
<b>Nombre de la Tarea:</b> Crear la Interfaz Gráfica mediante la cual se gestionan los contenidos y adicionan los mismos.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.3.
<b>Fecha Inicio:</b> 24/04/2012.	<b>Fecha Fin:</b> 26/04/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b> En esta Tarea se crea el grid necesario sobre el cual se lista la gestión de los contenidos.	

Tabla # 32 Tarea de Ingeniería # 1, HU # 10, Iteración # 4.

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

<b>Tarea de la Ingeniería</b>	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 11.
<b>Nombre de la Tarea:</b> Editar Contenido y Visualizar Cambios.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.3.
<b>Fecha Inicio:</b> 27/04/2012.	<b>Fecha Fin:</b> 28/04/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b> Se edita el contenido y se lista en el grid.	

Tabla # 33 Tarea de Ingeniería # 1, HU # 11, Iteración # 4.

<b>Tarea de la Ingeniería</b>	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 12.
<b>Nombre de la Tarea:</b> Eliminar Contenido y Visualizar.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.3.
<b>Fecha Inicio:</b> 29/04/2012.	<b>Fecha Fin:</b> 30/04/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b> Se elimina el contenido y se lista en el grid.	

Tabla # 34 Tarea de Ingeniería # 1, HU # 12, Iteración # 4.

### 4.3.4. Iteración # 5.

<b>Historias de Usuarios</b>	<b>Tiempo de Implementación</b>	
	<b>Estimación</b>	<b>Real</b>
Previsualizar Proyecto.	3	2

Tabla # 35 HUs Iteración # 5.

<b>Tarea de la Ingeniería</b>	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 13.
<b>Nombre de la Tarea:</b> Crear la Interfaz Gráfica mediante la cual se pre visualiza un proyecto.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 2.
<b>Fecha Inicio:</b> 01/05/2012.	<b>Fecha Fin:</b> 15/05/2012.

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

<b>Programador Responsable:</b> Adrian Bello Mesa.
<b>Descripción:</b> En esta Tarea se cargan todos los componentes pertenecientes a un proyecto determinado que existen en base de datos.

Tabla # 36 Tarea de Ingeniería # 1, HU # 13, Iteración # 5.

### 4.3.5. Iteración # 6.

Historias de Usuarios	Tiempo de Implementación	
	Estimación	Real
Exportar Proyecto.	3	2

Tabla # 37 HUs Iteración # 6.

Tarea de la Ingeniería	
Número de la Tarea: 1.	Número de la Historia de Usuario: 14.
Nombre de la Tarea: Exportar Proyecto.	
Tipo de Tarea: Desarrollo.	Puntos Estimados: 2.
Fecha Inicio: 16/05/2012.	Fecha Fin: 30/05/2012.
Programador Responsable: Adrian Bello Mesa.	
Descripción: Se guardan todos los componentes cargados en un directorio.	

Tabla # 38 Tarea de Ingeniería # 1, HU # 14, Iteración # 6.

### 4.3.6. Iteración # 7.

Historias de Usuarios	Tiempo de Implementación	
	Estimación	Real
Adicionar Glosario.	1	1
Editar Glosario.		
Eliminar Glosario.		
Autenticar		

Tabla # 39 HUs Iteración # 7.

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

Tarea de la Ingeniería	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 15.
<b>Nombre de la Tarea:</b> Crear la Interfaz Gráfica que permita visualizar y adicionar un glosario.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.25.
<b>Fecha Inicio:</b> 31/06/2012.	<b>Fecha Fin:</b> 01/06/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b>	

Tabla # 40 Tarea de Ingeniería # 1, HU # 15, Iteración # 7

Tarea de la Ingeniería	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 16.
<b>Nombre de la Tarea:</b> Editar un glosario y visualizar los cambios.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.25.
<b>Fecha Inicio:</b> 02/06/2012.	<b>Fecha Fin:</b> 03/06/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b> Editar un glosario y visualizarlo.	

Tabla # 41 Tarea de Ingeniería # 1, HU # 16, Iteración # 7.

Tarea de la Ingeniería	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 17.
<b>Nombre de la Tarea:</b> Eliminar un glosario y visualizar.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.25.
<b>Fecha Inicio:</b> 04/06/2012.	<b>Fecha Fin:</b> 05/06/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b>	

Tabla # 42 Tarea de Ingeniería # 1, HU # 17, Iteración # 7.

Tarea de la Ingeniería	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 18.

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

---

<b>Nombre de la Tarea:</b> Autenticar.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.25.
<b>Fecha Inicio:</b> 05/06/2012.	<b>Fecha Fin:</b> 06/06/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b> El usuario se autentica entrando su usuario y contraseña.	

Tabla # 43 Tarea de Ingeniería # 1, HU # 18, Iteración # 7.

### 4.4. Estándar de Codificación

En vistas a garantizar la legibilidad, facilidad de mantenimiento y reusabilidad del código escrito se definieron estándares de codificación para cada una de las categorías sintácticas usadas en la fase de implementación, díganse declaración de variables, declaración de funciones, entre otras.

#### Declaración de Clases

Se nombraron las Clases comenzando todas las palabras con mayúscula y unidas, previendo el uso de no más de tres palabras:

```
class GestionarAyudaContoller extends CI_Controller
```

#### Declaración de Funciones o Métodos:

Se utilizaron nombres que expresan literalmente el papel que cumplen en la confección de la aplicación, así como la Notación Camell Case y el uso de no más de tres palabras en los mismos:

```
public function getProyectos ()
```

#### Declaración de variables:

Se declararon las variables con el estándar de codificación de PHP y la Notación Camell Case a la hora de nombrarlas:

```
$idProyecto =
```



### **Identación:**

Se utilizó la indentación predeterminada por el IDE NetBeans a la hora de organizar el código y las sintaxis:

```
public function adicionarCategoria() {
    $titulo = $this->input->post('titulo');
    $proyecto = $this->input->post('proyecto');

    $this->load->model('GestionarAyudaModel');
    echo json_encode($this->GestionarAyudaModel->adicionarCategoria($titulo, $proyecto));
}
```

### **Comentarios:**

Se utilizó el comentario inicial de las clases como está predeterminado por el IDE Netbeans y a la hora de comentar una determinada línea o fragmento de código como se muestra:

```
/*Esto es un Comentario*/
```

### **4.5. Pruebas.**

Uno de los pilares fundamentales de XP es el proceso de pruebas, el cual constituye el último bastión desde el que se puede evaluar la calidad de forma pragmática y descubrir los errores. Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente. Por esta razón se debe definir en el proceso de la ingeniería del software. **(Pressman, 2005)**

La metodología XP divide las pruebas en dos grupos: pruebas unitarias, desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación, destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente.

#### **4.5.1. Pruebas Unitarias.**

Las pruebas unitarias según la metodología XP se utilizan para verificar el funcionamiento del sistema; se va escribiendo código y probándolo secuencialmente, HU por HU y tarea de ingeniería por tarea de

ingeniería, con el objetivo de garantizar calidad y funcionalidad desde los inicios del desarrollo, evitando la aparición de nuevos riesgos una vez terminada una iteración.

Estas pruebas son de gran importancia ya que se consideran fundamentales en el desarrollo de XP, pues los cambios son integrados en lapsos de tiempo reducidos. A su vez éstas brindan una visión de lo que se quiere realizar, demostrando que lo implementado es lo que se deseaba en la concepción de los requerimientos, permitiéndole al programador realizar cambios de forma segura respaldados por efectivos casos de prueba.

En el desarrollo de la solución planteada se utilizaron las facilidades del framework usado en el desarrollo de la aplicación. Éste consta de una clase diseñada para realizar las pruebas unitarias o de unidad llamada `unit_test`, formada por una función de evaluación y dos funciones de resultado. Este mecanismo permite evaluar el código y determinar si se está produciendo el tipo de dato y resultado correcto.

En este caso se utilizaron funciones brindadas por el framework diseñadas con el objetivo de evaluar los tipos de datos que se esperaba devolvieran las funciones codificadas, entre las cuales están:

- ✓ `is_string`
- ✓ `is_bool`
- ✓ `is_true`
- ✓ `is_false`
- ✓ `is_array`

Diseñando un caso de prueba global que permite evaluar cada una de las funciones de manera particular y generar una especie de reporte de pruebas. Este proceso dio como resultado lo expresado en la siguiente gráfica. (Ver Anexo # 6)

### **4.5.2. Pruebas de Aceptación.**

Las pruebas de aceptación son creadas en base a las HU, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una HU ha sido correctamente implementada. Las pruebas de aceptación son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una HU no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación.

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

---

Durante el proceso de pruebas se detectaron algunas no conformidades por parte del cliente que fueron subsanadas en la medida en que se iban culminando las iteraciones como propone la metodología. (Ver Anexo # 1)

### Casos de Pruebas

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU1_P1.	<b>HU:</b> 1.
<b>Nombre:</b> Crear Proyecto.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica de los proyectos que se creó el mismo con el nombre introducido por el cliente.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se creó el proyecto correctamente.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Crear Proyecto e introducir los datos.	
<b>Resultado Esperado:</b> El sistema debe mostrar el ítem creado con el nombre del proyecto introducido por el cliente.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla # 44 Caso de Prueba # 1 para HU Crear Proyecto.

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU2_P1.	<b>HU:</b> 2.
<b>Nombre:</b> Editar Proyecto.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica de los proyectos que se editó el mismo con el nombre introducido por el cliente y los cambios realizados.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se editó el proyecto correctamente.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Editar Proyecto e introducir los nuevos datos.	
<b>Resultado Esperado:</b> El sistema debe mostrar el ítem creado con los nuevos datos introducidos.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla # 45 Caso de Prueba # 1 para HU Editar Proyecto.

--

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU3_P1.	<b>HU:</b> 3.
<b>Nombre:</b> Eliminar Proyecto.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica de los proyectos que se eliminó el proyecto seleccionado con toda la información perteneciente al mismo.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se eliminó el proyecto correctamente.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Eliminar Proyecto.	
<b>Resultado Esperado:</b> El sistema debe eliminar de la lista el proyecto seleccionado con las categorías y fichas correspondientes.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla # 46 Caso de Prueba # 1 para HU Eliminar Proyecto

Se diseñan los Casos de Pruebas correspondientes a las HU #4, 5 y 6. (Ver Anexo # 3)

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU7_P1.	<b>HU:</b> 7.
<b>Nombre:</b> Adicionar Ficha.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica de las fichas que se adicionó la misma con el nombre introducido por el cliente una vez seleccionada la categoría correspondiente.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se adicionó la ficha correctamente	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Adicionar Ficha e introducir los datos.	
<b>Resultado Esperado:</b> El sistema debe mostrar la lista de ficha correspondiente a la categoría seleccionada y el ítem adicionado con el nombre introducido por el cliente.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla # 47 Caso de Prueba # 1 para HU Adicionar Ficha.

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU8_P1.	<b>HU:</b> 8.

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

<b>Nombre:</b> Editar Ficha.
<b>Descripción:</b> Comprobar mediante la interfaz gráfica de las fichas que se editó la misma con el nombre introducido por el cliente y los cambios realizados.
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se editó la ficha correctamente.
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Editar Ficha e introducir los nuevos datos.
<b>Resultado Esperado:</b> El sistema debe mostrar el ítem creado con los nuevos datos introducidos.
<b>Evaluación de la Prueba:</b> Satisfactoria.

Tabla # 48 Caso de Prueba # 1 para HU Editar Ficha.

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU9_P1.	<b>HU:</b> 9.
<b>Nombre:</b> Eliminar Ficha.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica de las fichas que se eliminó la ficha seleccionada con toda la información perteneciente a la misma.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se eliminó la ficha correctamente.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Eliminar Ficha.	
<b>Resultado Esperado:</b> El sistema debe eliminar de la categoría seleccionada la ficha deseada.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla # 49 Caso de Prueba # 1 para HU Eliminar Ficha.

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU10_P1.	<b>HU:</b> 10.
<b>Nombre:</b> Adicionar Contenido.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica de los contenidos que se adicionó el mismo con la configuración establecida por el cliente una vez seleccionada la ficha correspondiente.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se adicionó y configuró el contenido correctamente.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Adicionar Contenido y configurar los mismos (Imagen, Título y Contenido).	
<b>Resultado Esperado:</b> El sistema debe mostrar el ítem adicionado con el nombre del contenido y la	

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

configuración establecida por el cliente.

**Evaluación de la Prueba:** Satisfactoria.

Tabla # 50 Caso de Prueba # 1 para HU Adicionar Contenido.

### Caso de Prueba de Aceptación

**Código:** HU11\_P1.

**HU:** 11.

**Nombre:** Editar Contenido.

**Descripción:** Comprobar mediante la interfaz gráfica de los contenidos que se editó el mismo con el nombre y configuración establecido por el cliente y los cambios realizados.

**Condiciones de Ejecución:** El cliente debe de comprobar que se editó el contenido correctamente.

**Entrada / Pasos de Ejecución:** Acceder a la funcionalidad Editar Contenido e introducir los nuevos datos de configuración.

**Resultado Esperado:** El sistema debe mostrar el ítem creado con los nuevos datos introducidos.

**Evaluación de la Prueba:** Satisfactoria.

Tabla # 51 Caso de Prueba # 1 para HU Editar Contenido.

### Caso de Prueba de Aceptación

**Código:** HU12\_P1.

**HU:** 12.

**Nombre:** Eliminar Contenido.

**Descripción:** Comprobar mediante la interfaz gráfica de los contenidos que se eliminó el contenido seleccionado con toda la información perteneciente a al mismo.

**Condiciones de Ejecución:** El cliente debe de comprobar que se eliminó el contenido correctamente.

**Entrada / Pasos de Ejecución:** Acceder a la funcionalidad Eliminar Contenido.

**Resultado Esperado:** El sistema debe eliminar de la ficha seleccionada el contenido deseado.

**Evaluación de la Prueba:** Satisfactoria.

Tabla # 52 Caso de Prueba # 1 para HU Eliminar Contenido.

### Caso de Prueba de Aceptación

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

<b>Código:</b> HU13_P1.	<b>HU:</b> 13.
<b>Nombre:</b> Previsualizar Proyecto.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica generada por la aplicación para este fin si se hace una vista preliminar del proyecto seleccionado.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se genera una vista preliminar del proyecto seleccionado en la lista de proyectos disponible.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Previsualizar y seleccionar el proyecto deseado de la lista disponible.	
<b>Resultado Esperado:</b> El sistema debe mostrar una vista preliminar del proyecto seleccionado en la lista de proyectos una vez accedido a esta funcionalidad.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla # 53 Caso de Prueba # 1 para HU Previsualizar Proyecto.

### Caso de Prueba de Aceptación

<b>Código:</b> HU14_P1.	<b>HU:</b> 14.
<b>Nombre:</b> Exportar Proyecto.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica generada por la aplicación para este fin si se guarda en el directorio predeterminado la página Web generada perteneciente al proyecto seleccionado.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se genera una página Web perteneciente al proyecto seleccionado con anterioridad.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Exportar y seleccionar el proyecto deseado de la lista disponible y localizar la página Web generada en el directorio predeterminado para este fin.	
<b>Resultado Esperado:</b> El sistema debe generar una página Web y guardarla en un directorio predeterminado con toda la información configurada por el usuario.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla # 54 Caso de Prueba # 1 para HU Exportar Proyecto.

### Caso de Prueba de Aceptación

<b>Código:</b> HU15_P1.	<b>HU:</b> 15.
-------------------------	----------------

## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

<b>Nombre:</b> Adicionar Glosario.
<b>Descripción:</b> Comprobar mediante la interfaz gráfica del glosario que se adicionó el mismo con la información introducida por el cliente (palabra y significado).
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se adicionó el glosario correctamente.
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Adicionar Glosario e introducir las palabras y su significado.
<b>Resultado Esperado:</b> El sistema debe mostrar el componente adicionado característico del glosario y el contenido introducido por el cliente.
<b>Evaluación de la Prueba:</b> Satisfactoria.

Tabla # 55 Caso de Prueba # 1 para HU Adicionar Glosario.

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU16_P1.	<b>HU:</b> 16.
<b>Nombre:</b> Editar Glosario.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica definida para el glosario que se editó el contenido del glosario en cuestión y se mostraron sus cambios.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se editó el glosario correctamente.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Editar Glosario e introducir los nuevos datos.	
<b>Resultado Esperado:</b> El sistema debe mostrar el componente creado con los nuevos datos introducidos.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla # 56 Caso de Prueba # 1 para HU Editar Glosario.

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU17_P1.	<b>HU:</b> 17.
<b>Nombre:</b> Eliminar Glosario.	
<b>Descripción:</b> Comprobar mediante la interfaz definida para ello que se eliminó el glosario confeccionado con anterioridad.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se eliminó el glosario correctamente.	



## CAPÍTULO 4: Desarrollo y Prueba de la Solución Propuesta

<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Eliminar Glosario una vez seleccionado el proyecto deseado.
<b>Resultado Esperado:</b> El sistema debe eliminar el glosario perteneciente al proyecto seleccionado.
<b>Evaluación de la Prueba:</b> Satisfactoria.

Tabla # 57 Caso de Prueba # 1 para HU Eliminar Glosario.

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU18_P1.	<b>HU:</b> 18.
<b>Nombre:</b> Autenticar.	
<b>Descripción:</b> Comprobar mediante la interfaz definida para ello que entraron los datos correctos y se accedió a la aplicación.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se autenticó correctamente.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la aplicación con los datos correctos mediante la autenticación.	
<b>Resultado Esperado:</b> El sistema debe dar acceso a las funcionalidades del mismo si se introdujeron los datos de autenticación correctos.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla # 58 Caso de Prueba # 1 para HU Autenticar.

### 4.6. Conclusiones.

La metodología aplicada es una de las más usadas y exitosas en cuanto a metodologías de desarrollo ágiles se refiere. Durante el desarrollo de este capítulo y el de la aplicación misma se pudo evidenciar la estabilidad y robustez que brinda XP en combinación con el análisis, diseño y arquitectura propuestos. Teniendo en cuenta la manera organizada en que se llevan a cabo las últimas dos fases del ciclo de vida, se garantiza la generación de un código organizado y limpio de errores y redundancias sintácticas. Permitiendo además el cumplimiento estricto de las necesidades del cliente gracias a su participación constante y orientadora durante todo el proceso de desarrollo de la aplicación. Avalando todo lo antes mencionado con la realización de las pruebas de calidad descritas en este capítulo, concluyendo así lo planificado y estimado en apartados anteriores.

# CONCLUSIONES

---

## CONCLUSIONES

Una vez finalizado el proceso de desarrollo del tema abarcado y la investigación generada para soportar la puesta en práctica e implementación de una Plataforma de Gestión de Ayudas, se puede llegar a las siguientes conclusiones.

- ✓ Se consideran cumplidos los objetivos trazados en el marco teórico de la investigación, así como automatización del proceso descrito por el cliente según sus necesidades.
- ✓ El desarrollo de la aplicación basada e implementada totalmente sobre el uso y aprovechamiento de herramientas libres, permitió justificar y explotar los beneficios del software libre en el desarrollo de la investigación, así como el uso sin limitaciones privativas de la solución propuesta.
- ✓ Teniendo en cuenta la metodología de desarrollo usada durante todo el proceso, se puede afirmar la importancia que tiene el uso de XP en el desarrollo de soluciones informáticas estables, en la implementación de proyectos de pequeño y mediano alcance.
- ✓ Como principal resultado y motivo de la investigación se obtuvo una Plataforma de Gestión de Ayudas para Sistemas de Información Geográficas en entornos Web garantizando:
  - El desarrollo de Sistemas de Ayudas de forma ágil y fácil.
  - El uso y reutilización del código generado en otras aplicaciones o en la implementación de nuevas versiones del software.
  - El ahorro en tiempo y recurso tanto informáticos como humanos a la hora de desarrollar ayudas para SIG.
  - La estandarización del proceso de desarrollo de Ayudas para SIG.
  - La independencia que brinda el software de no solo implementar ayudas para SIG, sino también para cualquier otra aplicación Web que sea necesario por el carácter genérico y estándar que brinda la misma.

# RECOMENDACIONES

---

## RECOMENDACIONES

Desarrollar un Sistema Gestor de Ayudas que cumpla con las características funcionales de un sistema de ayuda inteligente, que vincule la inteligencia artificial en su desarrollo y que además proporcione al desarrollador la posibilidad de adaptabilidad y cotejamiento con la interfaz visual del SIG para el cual esté diseñada la aplicación generada.

Implementar funcionalidades de vistas de contenido dinámica, de manera que se logre interactividad directa entre los aplicativos y la ayuda generada.

Incluir en la plataforma una biblioteca de íconos estándares que representen funcionalidades comunes de los SIG.

Utilizar la plataforma para la generación de ayudas de otros productos que no sean SIG, pero que cumplan con los parámetros de ser aplicaciones Web.

## BIBLIOGRAFÍA CONSULTADA

---

### BIBLIOGRAFÍA CONSULTADA

1. Álvarez, Miguel Angel. 2007. *Manual de PHP 5* [Online]2007  
<http://www.desarrolloweb.com/manuales/58/>
2. Alvarez, Will A. 2008. *Funadamentos Básicos de PHP+HTML*.
3. Beck, Kent. 2005. *Extreme Programming Explained: Embrace Change, 2nd Edition*.
4. Canales, Roberto. 2004. *Patrones GRASP*.
5. Canós, José H. 2010. *Metodologías de Desarrollo ágiles.SCRUM y XP*. [Online]2010  
<http://www.slideshare.net/ejordi/metodologas-de-desarrollo-giles-scrum-xp>
6. Carroll, Aaronson. 1988. *Learning by doing with simulated intelligent help*.
7. Calero, Manuel. 2003. *Una explicación de la programación extrema (XP)*. [Online]2003  
<http://www.apolosoftware.com/>
8. Cerda, Felipe. 2009. *El único IDE que necesitas*. [Online]2009  
<http://www.slideshare.net/felipecerda/netbeans-el-nico-ide-que-necesitas>
9. Chang, Kai Chen.2006.*Introduction to Geographic Information Systems*.
10. Cigarrán, Fernandez-Manjon y Buenaga. 1996. *El sistema Argos. Departamento de Informática y Automática. UCM*.
11. Duffy, Mehlenbacher, Palmer.1989. *The evaluation of online help systems*.
12. Duffy, Mehlenbacher, Palmer.1992. *On Line Help Design and Evaluation*.
13. ESRI Europa. 2004.*Tendencias de los Sistemas de Información Geográfica* [Online]2004  
[http://www.elagrimensor.net/elearning/lecturas/Trend\\_SIG1.pdf](http://www.elagrimensor.net/elearning/lecturas/Trend_SIG1.pdf)
14. Fabien Potencier.2008. *Breve Historia de PostgreSQL*. [Online]2012  
<http://todosloscursosu.blogspot.com/2012/03/breve-historia-de-postgres.html>
15. Kearsley, Greg.1987. *Artificial Intelligence & Instruction*.
16. Kearsley, Greg. 1988. *Online Help System: Design and Implementation*.
17. Letelier, Patricio y Penadés, Carmen. 2003. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*.
18. Lobo, Luis. 2006. *Javascript Extendido Ext JS*.
19. Lozano, Víktor. 2006. *Manual de Code Igniter en Español*.
20. Microsoft DeveloperNetworkhttp (MSDN) .2010. *Sistemas de ayuda en aplicaciones de Windows*[Online] 2010[http://msdn.microsoft.com/es-es/library/55dct0a1\(VS.85\).aspx](http://msdn.microsoft.com/es-es/library/55dct0a1(VS.85).aspx)
21. Morales, .Miguel.2008. *Arquitectura de Software*.
22. Öztürk, Ahmet. 2002. *Computing Information Services Newsletters*. [Online] 2002.  
<http://cisin.metu.edu.tr/2002-6/free.php>

## BIBLIOGRAFÍA CONSULTADA

---

23. Pressman, Roger. 2005. *Introducción a la Ingeniería de Software*.
24. Ramírez, Alejandro y Edison, Agrim. 2009. *Implementación de un Sistema de Información Geográfica con un perfil aplicable al sector arrocero*. [Online] 2009. [http://egal2009.easyplanners.info/area04/4103\\_Ramirez\\_Cardozo\\_Alejandro.pdf](http://egal2009.easyplanners.info/area04/4103_Ramirez_Cardozo_Alejandro.pdf)
25. Sanchez, Emilio A. 2004. *Mejorando le gestión de historias de usuario en Extreme Programming*. [Online] 2004. <http://issi.dsic.upv.es/publications/archives/f-1068918773758/JISBD03SanchezLetelier.pdf>
26. Shank. 1981. *Where is the AI? AI Magazine*.
27. Wilensky, Chin. 1989. *The Berkeley UNIX Consultant Project*.
28. Winkels, Breuker. 1992. *Wath is a ITS?New directions for intelligent Tutoring System*.

# BIBLIOGRAFÍA CITADA

---

## BIBLIOGRAFÍA CITADA

1. Adobe Product. 2012. Robohelp [Online]2012  
<http://www.images.adobe.com/www.adobe.com/content/dam/Adobe/en/products/robohelp/pdf/us-robohelp-ue-fnl-09152011.pdf>
2. Álvarez, Miguel Angel. 2007. Manual de PHP 5 [Online]2007  
<http://www.desarrolloweb.com/manuales/58/>
3. Alvarez, Will A. 2008. *Funadamentos Básicos de PHP+HTML*.
4. Beck, Kent. 2005. *Extreme Programming Explained: Embrace Change, 2nd Edition*.
5. Canales, Roberto. 2004. *Patrones GRASP*.
6. Canós, José H. 2010. *Metodologías de Desarrollo ágiles.SCRUM y XP*. [Online]2010  
<http://www.slideshare.net/ejordi/metodologas-de-desarrollo-giles-scrum-xp>
7. Carroll, Aaronson. 1988. *Learning by doing with simulated intelligent help*.
8. Calero, Manuel. 2003. *Una explicación de la programación extrema (XP)*. [Online]2003  
<http://www.apolosoftware.com/>
9. Cerda, Felipe. 2009. *El único IDE que necesitas*. [Online]2009  
<http://www.slideshare.net/felipecerda/netbeans-el-nico-ide-que-necesitas>
10. Cinfonet. 2005. *Apache Web Server*. [Online]2005.  
<http://hospedaje.cinfonet.com/?web=apache>
11. Chang, Kai Chen.2006.*Introduction to Geographic Information Systems*.
12. Cigarrán, Fernandez-Manjon y Buenaga. 1996. *El sistema Argos. Departamento de Informática y Automática. UCM*.
13. Duffy, Mehlenbacher, Palmer.1989. *The evaluation of online help systems*.
14. Duffy, Mehlenbacher, Palmer.1992. *On Line Help Design and Evaluation*.
15. ESRI Europa. 2004.*Tendencias de los Sistemas de Información Geográfica* [Online]2004  
[http://www.elagrimensor.net/elearning/lecturas/Trend\\_SIG1.pdf](http://www.elagrimensor.net/elearning/lecturas/Trend_SIG1.pdf)
16. Fabien Potencier.2008. *Breve Historia de PostgreSQL*. [Online]2012  
<http://todosloscursosu.blogspot.com/2012/03/breve-historia-de-postgres.html>
17. Kearsley, Greg.1987. *Artificial Intelligence & Instruction*.
18. Kearsley, Greg. 1988. *Online Help System: Design and Implementation*.
19. Letelier, Patricio y Penadés, Carmen. 2003. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*.
20. Lobo, Luis. 2006. *Javascript Extendido Ext JS*.

## BIBLIOGRAFÍA CITADA

---

21. Lozano, Víktor. 2006. *Manual de Code Igniter en Español*.
22. Microsoft DeveloperNetworkhttp (MSDN) .2010. *Sistemas de ayuda en aplicaciones de Windows*[Online] 2010[http://msdn.microsoft.com/es-es/library/55dct0a1\(VS.85\).aspx](http://msdn.microsoft.com/es-es/library/55dct0a1(VS.85).aspx)
23. Morales, .Miguel.2008. *Arquitectura de Software*.
24. Öztürk, Ahmet. 2002. *Computing Information Services Newsletters*. [Online] 2002. <http://cisin.metu.edu.tr/2002-6/free.php>
25. Pressman, Roger. 2005. *Introducción a la Ingeniería de Software*.
26. Ramírez, Alejandro y Edison, Agrim. 2009. *Implementación de un Sistema de Información Geográfica con un perfil aplicable al sector arrocero*. [Online] 2009. [http://egal2009.easyplanners.info/area04/4103\\_Ramirez\\_Cardozo\\_Alejandro.pdf](http://egal2009.easyplanners.info/area04/4103_Ramirez_Cardozo_Alejandro.pdf)
27. Sanchez, Emilio A. 2004. *Mejorando le gestión de historias de usuario en Extreme Programming*. [Online] 2004. <http://issi.dsic.upv.es/publications/archives/f-1068918773758/JISBD03SanchezLetelier.pdf>
28. Shank. 1981. *Where is the AI? AI Magazine*.
29. Softlockup. 2009. *HelpMagician*. [Online] 2009. <http://www.softlookup.com/display.asp?id=7919>
30. Wilensky, Chin. 1989. *The Berkeley UNIX Consultant Project*.
31. Winkels, Breuker. 1992. *Wath is a ITS?New directions for intelligent Tutoring System*.
32. Microsoft Corporation. 2012 *Winhelp*. [Online]2012 <http://msdn.microsoft.com/es-es/library/cc466485%28v=vs.71%29.aspx>

# ANEXOS

---

## ANEXOS

### Anexo # 1

#### Historias de Usuario # 4,5 y 6.

Historia de Usuario	
<b>Número:</b> 4.	<b>Nombre de Historia:</b> Adicionar Categoría.
<b>Usuario:</b> Cliente o Desarrollador.	
<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo en Desarrollo:</b> Medio.
<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 2.
<b>Descripción:</b> El cliente selecciona un proyecto X y el sistema brinda la posibilidad de adicionarle una categoría y lo visualiza.	
<b>Observaciones:</b> El cliente selecciona el proyecto (X) antes creado y el sistema brinda la posibilidad de adicionarle una categoría (Y) donde ésta va a formar parte del árbol jerárquico donde (Y) categoría va a ser un subnodo de (X) proyecto.	

Tabla # 59 HU Adicionar Categoría.

Historia de Usuario	
<b>Número:</b> 5.	<b>Nombre de Historia:</b> Editar Categoría.
<b>Usuario:</b> Cliente o Desarrollador.	
<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo en Desarrollo:</b> Medio.
<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 2.
<b>Descripción:</b> El cliente selecciona una categoría (Y), y el sistema brinda la posibilidad de editarla y lo visualiza.	



## ANEXOS

**Observaciones:** El cliente selecciona el proyecto (X) antes creado y el sistema brinda la posibilidad de seleccionar una de sus (Y) categorías y editarlas guardando los cambios realizados y lo visualiza.

Tabla # 60 HU Editar Categoría.

<b>Historia de Usuario</b>	
<b>Número:</b> 6.	<b>Nombre de Historia:</b> Eliminar Categoría.
<b>Usuario:</b> Cliente o Desarrollador.	
<b>Prioridad en el Negocio:</b> Alta.	<b>Riesgo en Desarrollo:</b> Medio.
<b>Puntos de Estimación:</b> 1.	<b>Iteración Asignada:</b> 2.
<b>Descripción:</b> El cliente selecciona una categoría (Y), y el sistema brinda la posibilidad de eliminarla y lo visualiza.	
<b>Observaciones:</b> El cliente selecciona el proyecto (X) antes creado y el sistema brinda la posibilidad de seleccionar una de sus (Y) categorías y eliminarla junto con sus hijos, del árbol jerárquico que conforma y lo visualiza.	

Tabla # 61 HU Eliminar Categoría.

### Anexo # 2

#### Tareas de Ingeniería de la Iteración # 2.

<b>Historias de Usuarios</b>	<b>Tiempo de Implementación</b>	
	<b>Estimación</b>	<b>Real</b>
Adicionar Categoría.	1	1
Editar Categoría.		
Eliminar Categoría.		

Tabla # 62 HUs Iteración # 2

<b>Tarea de la Ingeniería</b>
-------------------------------

## ANEXOS

<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 4.
<b>Nombre de la Tarea:</b> Crear la Interfaz Gráfica mediante la cual se gestionan las categorías y adicionan las mismas.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.3.
<b>Fecha Inicio:</b> 08/04/2012.	<b>Fecha Fin:</b> 11/04/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b> En esta Tarea se crea el grid necesario sobre el cual se lista la gestión de las Categorías.	

Tabla # 63 Tarea de Ingeniería # 1, HU # 4, Iteración # 2.

<b>Tarea de la Ingeniería</b>	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 5.
<b>Nombre de la Tarea:</b> Editar Categoría y Visualizar Cambios.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.3.
<b>Fecha Inicio:</b> 12/04/2012.	<b>Fecha Fin:</b> 13/04/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b> Se edita la Categoría y se lista en el grid.	

Tabla # 64 Tarea de Ingeniería # 1, HU # 5, Iteración # 2.

<b>Tarea de la Ingeniería</b>	
<b>Número de la Tarea:</b> 1.	<b>Número de la Historia de Usuario:</b> 6.
<b>Nombre de la Tarea:</b> Eliminar Categoría y Visualizar.	
<b>Tipo de Tarea:</b> Desarrollo.	<b>Puntos Estimados:</b> 0.3.
<b>Fecha Inicio:</b> 14/04/2012.	<b>Fecha Fin:</b> 15/04/2012.
<b>Programador Responsable:</b> Adrian Bello Mesa.	
<b>Descripción:</b> Se elimina la categoría y se lista en el grid.	

Tabla # 65 Tarea de Ingeniería # 1, HU # 5, Iteración # 2.

# ANEXOS

## Anexo # 3

### Casos de Prueba de las HU # 4, 5 y 6.

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU4_P1.	<b>HU:</b> 4.
<b>Nombre:</b> Adicionar Categoría.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica de las categorías que se adicionó la misma con el nombre introducido por el cliente una vez seleccionado el proyecto correspondiente.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se adicionó la categoría. correctamente	
<b>Entrada / Pasos de Ejecución:</b> Seleccionar el proyecto y Acceder a la funcionalidad Adicionar Categoría e introducir los datos.	
<b>Resultado Esperado:</b> El sistema debe mostrar las categorías del proyecto seleccionado y el ítem adicionado con el nombre de la categoría introducida por el cliente.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla # 66 Caso de Prueba # 1 para HU Adicionar Categoría.

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU5_P1.	<b>HU:</b> 5.
<b>Nombre:</b> Editar Categoría.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica de las categorías que se editó la misma con el nombre introducido por el cliente y los cambios realizados.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se editó la categoría correctamente.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Editar Categoría e introducir los nuevos datos.	
<b>Resultado Esperado:</b> El sistema debe mostrar el ítem creado con los nuevos datos introducidos.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla # 67 Caso de Prueba # 1 para HU Editar Categoría.

--

## ANEXOS

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU6_P1.	<b>HU:</b> 6.
<b>Nombre:</b> Eliminar Categoría.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica de las categorías que se eliminó la categoría seleccionada con toda la información perteneciente a la misma.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se eliminó la categoría correctamente.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Eliminar Categoría.	
<b>Resultado Esperado:</b> El sistema debe eliminar de la lista de categorías del proyecto seleccionado la categoría seleccionada con todas fichas pertenecientes a la misma	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla # 68 Caso de Prueba # 1 para HU Eliminar Categoría.

### Anexo # 4

#### No Conformidades

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU4_P1.	<b>HU:</b> 4.
<b>Nombre:</b> Adicionar Categoría.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica de las categorías que se adicionó la misma con el nombre introducido por el cliente una vez seleccionado el proyecto correspondiente.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se adicionó la categoría correctamente.	
<b>Entrada / Pasos de Ejecución:</b> Seleccionar el proyecto y Acceder a la funcionalidad Adicionar Categoría e introducir los datos.	
<b>Resultado Esperado:</b> El sistema debe mostrar las categorías del proyecto seleccionado y el ítem adicionado con el nombre de la categoría introducida por el cliente.	
<b>Evaluación de la Prueba:</b> No Satisfactoria	

Tabla # 69 Caso de Prueba # 1 para HU Adicionar Categoría con Conflicto.

No conformidad detectada: El sistema muestra las categorías de todos los proyectos existentes y no las del proyecto seleccionado particularmente.

## ANEXOS

---

Caso de Prueba de Aceptación	
<b>Código:</b> HU6_P1.	<b>HU:</b> 6.
<b>Nombre:</b> Eliminar Categoría.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica de las categorías que se eliminó la categoría seleccionada con toda la información perteneciente a la misma.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se eliminó la categoría correctamente.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Eliminar Categoría.	
<b>Resultado Esperado:</b> El sistema debe eliminar de la lista de categorías del proyecto seleccionado la categoría seleccionada con todas fichas pertenecientes a la misma.	
<b>Evaluación de la Prueba:</b> Satisfactoria.	

Tabla # 70 Caso de Prueba # 1 para HU Eliminar Categoría con Conflicto.

No conformidad detectada:

El sistema no elimina las fichas pertenecientes a la categoría seleccionada.

Caso de Prueba de Aceptación	
<b>Código:</b> HU10_P1.	<b>HU:</b> 10.
<b>Nombre:</b> Adicionar Contenido.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica de los contenidos que se adicionó el mismo con la configuración establecida por el cliente una vez seleccionada la ficha correspondiente.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se adicionó y configuró el contenido correctamente.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Adicionar Contenido y configurar los contenidos (Imagen, Título y Contenido).	
<b>Resultado Esperado:</b> El sistema debe mostrar el ítem adicionado con el nombre del contenido y la configuración establecida por el cliente.	
<b>Evaluación de la Prueba:</b> No Satisfactoria.	

Tabla # 71 Caso de Prueba # 1 para HU Adicionar Contenido con Conflicto.

## ANEXOS

---

### No conformidad detectada:

El sistema no permite configurar la existencia o no de imagen en el contenido de la ficha seleccionada, pues obliga el uso de imágenes en el mismo.

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b> HU14_P1.	<b>HU:</b> 14.
<b>Nombre:</b> Exportar Proyecto.	
<b>Descripción:</b> Comprobar mediante la interfaz gráfica generada por la aplicación para este fin si se guarda en el directorio predeterminado la página Web generada perteneciente al proyecto seleccionado.	
<b>Condiciones de Ejecución:</b> El cliente debe de comprobar que se genera una página Web perteneciente al proyecto seleccionado con anterioridad.	
<b>Entrada / Pasos de Ejecución:</b> Acceder a la funcionalidad Exportar y seleccionar el proyecto deseado de la lista disponible y localizar la página Web generada en el directorio predeterminado para este fin.	
<b>Resultado Esperado:</b> El sistema debe generar una página Web y guardarla en un directorio predeterminado por el sistema con toda la información configurada por el usuario.	
<b>Evaluación de la Prueba:</b> No Satisfactoria.	

Tabla # 72 Caso de Prueba # 1 para HU Exportar Proyecto con Conflicto

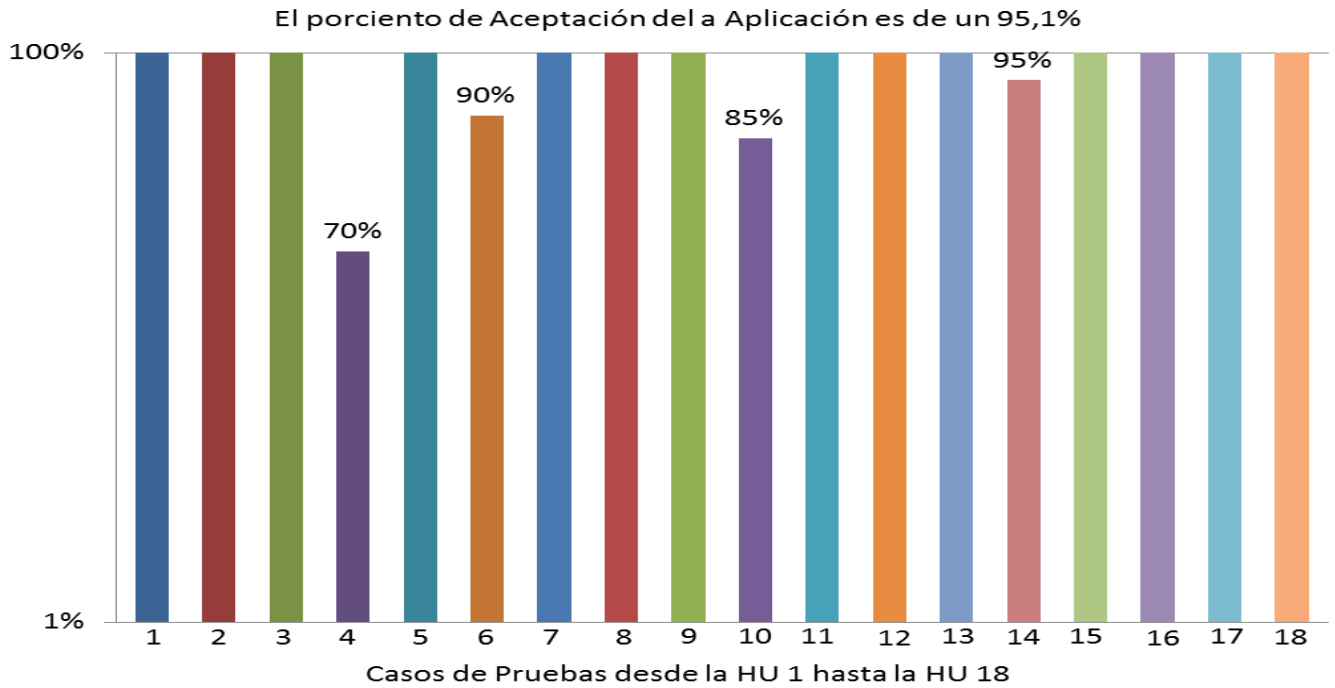
### No conformidad detectada:

El sistema genera una página Web que no especifica el nombre del proyecto o del SIG para la cual está creada.

# ANEXOS

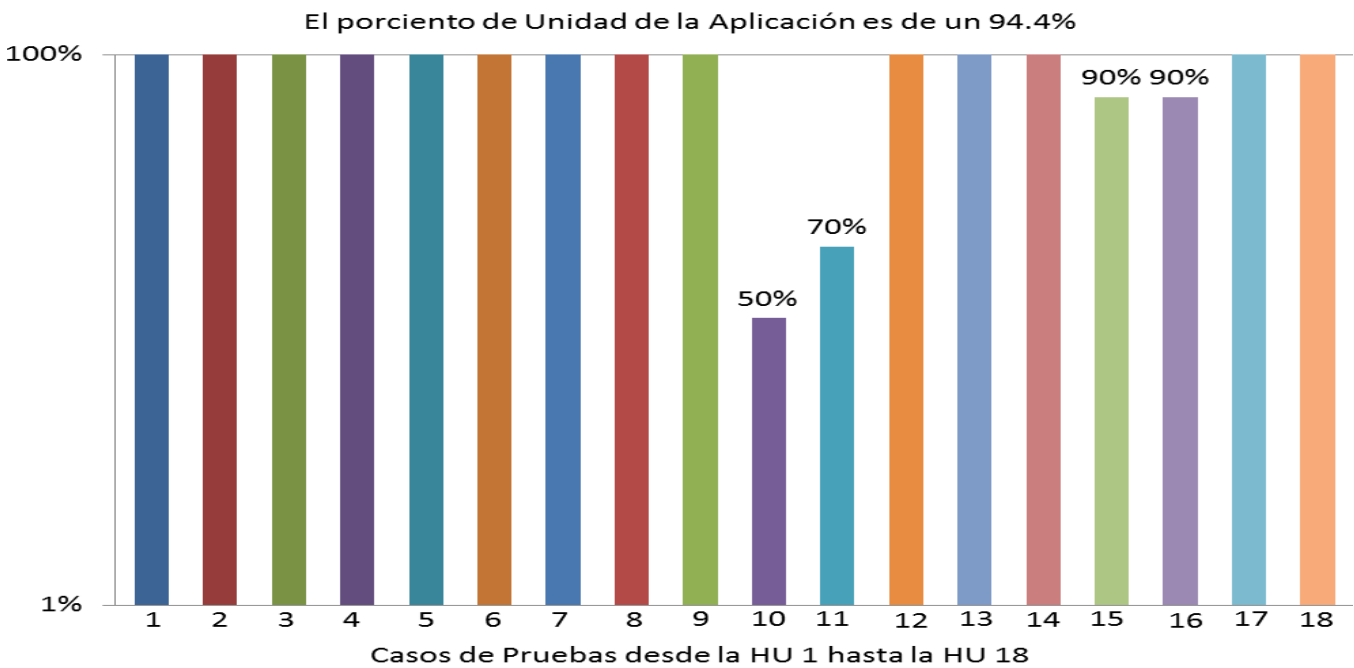
## Anexo # 5

### Resultados de Pruebas de Aceptación



## Anexo # 6

### Resultados de Pruebas de Unidad o Unitarias



## No conformidades detectadas:

Los casos de prueba Adicionar y Editar Contenido (10 y 11 respectivamente), sugirieron cambiar la manipulación de las imágenes introducidas en el sistema, del uso de base de datos a archivos, gestionando las mismas por su nombre y no por la imagen en específico.

Los casos de prueba Adicionar y Editar Glosario (15 y 16 respectivamente), sugirieron el uso del tipo de dato VARCHAR a TEXT para evitar conflictos con la cantidad de caracteres permitidos por cada tipo de dato.

## **Anexo # 7**

### **Diseño de Base de Datos**

